

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE

وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique

جامعة أبي بكر بلقايد - تلمسان -
Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du diplôme de **MASTER**

En : (Automatique)

Spécialité : Automatique et Informatique Industrielle

Par :

**AFFANE Meriem
BENALI MEDJAHED Fatima Zohra**

Sujet

**Classification des gestes chirurgicaux par
apprentissage automatique supervisé**

Soutenu publiquement, le : 24 / 06 / 2026 , devant le jury composé de :

Mr.HADJ ABDELKADER Amine	Med	Professeur	(Univ. Tlemcen)	Président
Mr. BENARIBA Hassane		MCA	(Univ. Tlemcen)	Examineur
Mme. SAIDI Farah		MCB	(Univ. Tlemcen)	Encadreur
Mr. BENMANSOUR Malik		Docteur	(Univ. Tlemcen)	Co-Encadreur

Année universitaire : 2025/2026

Remerciements

Louange à Dieu le Tout-Puissant, Clément et Miséricordieux, qui nous a guidés tout au long de cette modeste réalisation.

Nous aimerions adresser nos plus sincères remerciements à notre encadrant, **Dr.SAIDI Farah**, pour sa disponibilité constante et ses conseils judicieux. Sa guidance précieuse a été d'une importance capitale dans la réalisation de ce travail.

Nous remercions vivement notre co-encadrant, **Dr.Malik BENMANSOUR**, pour sa disponibilité et ses précieux conseils tout au long de ce PFE.

Nous tenons à remercier particulièrement **Pr.HADJ ABDELKADER Mohamed Amine**, président du jury, **Dr.BENARIBA Hassane**, examinateur, pour le temps consacré à l'évaluation de ce mémoire, leurs remarques constructives et leurs contributions importantes pour perfectionner notre travail.

Nous souhaitons exprimer notre profonde gratitude envers l'ensemble des enseignants de la FT (Faculté de Technologie) qui ont contribué à notre formation, en particulier les enseignants du département de Génie Electrique et Electronique.

Enfin, nous tenons à remercier notre deuxième famille et nos amis de la FT, qui ont été nombreux à nous aider et à nous encourager tout au long de notre parcours universitaire. Nous sommes sincèrement reconnaissants envers toutes les personnes mentionnées ci-dessus pour leur soutien, leurs conseils et leur contribution à la réalisation de ce mémoire.

Dédicace

Ma dédicace s'adresse premièrement et avant tout à la plus belle créature que Dieu a créé sur terre, ma source de tendresse, de patience et de générosité

Ma mère !

*À ce qui est la source de mon inspiration et mon courage, à qui je dois de
l'amour,
de la reconnaissance et du respect*

Mon père !

À mes cheres Soueurs Assia et Sarah

*À tous qui non Ceux qui n'ont pas cessé de me diriger par ses précieux
conseils,*

À mes amis proches sans exception surtout Marwa, Wissem, Samaa Eddounia

À mon amie intime et mon binôme aussi Fatima Zohra

AFFANE Meriem

Dedicace

Je dédie ce travail de fin d'études à mes parents, dont le soutien indéfectible et les encouragements constants ont été une source d'inspiration tout au long de ce parcours.

Leur amour et leur dévouement m'ont permis de surmonter les défis et de persévérer jusqu'à l'aboutissement de ce projet.

À mes soeurs Aicha, Imen, Chahinez, Hidayet, Batoul, et mon frère Abdelkader pour leur soutien et leurs encouragements continus. Vous avez toujours cru en moi, même dans les moments les plus difficiles.

À ma copine MALITI Meriem, pour sa patience et son amour. Ton soutien moral a été inestimable et m'a aidé à garder le cap.

À tous mes amis, pour les moments de partage, de joie et de motivation.

Merci à vous tous pour votre soutien. Ce travail est le fruit de notre effort collectif et de notre solidarité.

À mon amie intime et mon binôme AFFANE Meriem

Table des matières

Remerciements	1
Dédicace	2
Dédicace	3
Introduction générale	8
1 État de l’art	10
1.1 Introduction	10
1.2 Chirurgie et gestes chirurgicaux	10
1.2.1 Définition	10
1.2.2 Les différentes méthodes de reconnaissance des gestes	11
1.3 Intelligence artificielle	12
1.3.1 Définition	12
1.3.2 Aspect aux techniques de l’intelligence artificielle	12
1.4 Apprentissage automatique	13
1.4.1 Définition	13
1.4.2 Types	13
1.4.3 Méthodes basées sur la classification des gestes chirurgicaux	15
1.5 Conclusion	17
2 Fondements théoriques des modèles hybrides CNN-LSTM	18
2.1 Réseaux de Neurones Convolutifs	18
2.1.1 Définition	18
2.1.2 Composants de base d’un CNN	19
2.1.3 Différentes Architecture de CNN	23
2.2 Long Short-Term Memory (LSTM)	25
2.2.1 Définition	25
2.2.2 Architecture et le mécanisme du LSTM	25
2.3 Modèle hybride CNN-LSTM	27
2.3.1 Définition	27
2.3.2 Architecture Hybride CNN-LSTM	28
2.3.3 Évaluation du modèle	28
2.3.4 Mécanisme d’intégration hybride de base	29
2.4 Conclusion	30
3 Expérimentations et résultats du modèle hybride CNN-LSTM sur la base de données JIGSAWS	31
3.1 Présentation de la base de données JIGSAWS	31
3.1.1 Description générale de JIGSAWS	31

3.1.2	Les trois gestes chirurgicaux étudiés	32
3.1.3	Prétraitement des données	32
3.2	Protocole expérimental et paramétrage du modèle	32
3.2.1	Architecture du modèle CNN-LSTM retenue	32
3.2.2	Configuration des hyperparamètres	33
3.2.3	Stratégie de validation	34
3.3	Simulation	35
3.3.1	Environnement logiciel et matériel	35
3.3.2	Résultats des 3 gestes	35
3.4	conclusion	43
Conclusion générale		44
Résumé		45

Liste Des Figures

1.1	Exemples de surgèmes dans des environnements chirurgicaux réels et sur fantômes : (a) positionnement de la pointe de l'aiguille sur le point d'insertion, (b) passage de l'aiguille à travers le tissu, (c) retrait de l'aiguille hors du tissu [1].	10
1.2	Les composantes de l'intelligence artificielle	13
1.3	Architecture du modèle proposé avec ses trois composantes (spatiale, temporelle et segmentaire) et focus sur la composante spatiale[2].	15
1.4	Architecture du réseau de neurones convolutif VGG[3].	16
2.1	Le pipeline d'un réseau de neurones convolutionnel (CNN)[4]	19
2.2	Schéma du processus convolution [5].	20
2.3	Calcul des valeurs de sortie d'une opération de max pooling 3×3 sur une entrée 5×5 [4].	21
2.4	La structure générale des fonctions d'activation[4].	22
2.5	Diagramme des différentes fonctions d'activation [4].	22
2.6	L'architecture de la cellule LSTM[6]	27
2.7	Architecture hybride CNN-LSTM[7]	28
3.1	Courbes d'évolution de la perte (loss) et de l'exactitude (accuracy)	36
3.2	Courbes d'évolution de la perte (loss) et de l'exactitude (accuracy) sur les ensembles d'entraînement et de test	36
3.3	Courbes d'apprentissage et performances maximales du modèle	37
3.4	Comparaison globale des courbes d'apprentissage et de performance (Entraînement, Validation et Test)	37
3.5	Répartition des vidéos par classe dans le jeu de test	38
3.6	Exactitude du modèle par classe de geste sur le jeu de test	39
3.7	Métriques de performance (Précision, Rappel, F1-Score) par classe sur le jeu de test.	40
3.8	Matrice de confusion brute du modèle sur l'ensemble de test	41
3.9	Rapport global des performances de classification du modèle sur l'ensemble de test	42
3.10	Rapport global des performances de classification du modèle sur l'ensemble de test	43

Liste Des tableau

- 2.1 Comparaison entre différentes architectures de CNN populaires [8] 23
- 3.1 la structure interne de réseau hybride CNN-LSTM 33

Introduction générale

L'intelligence artificielle et l'apprentissage automatique connaissent une expansion rapide et deviennent des moteurs clés de l'innovation dans les domaines scientifiques et médicaux. En chirurgie moderne, notamment la chirurgie assistée par robot, l'analyse numérique du déroulement des interventions ouvre des perspectives inédites pour la formation des praticiens. Jusqu'ici, l'évaluation des compétences des chirurgiens en formation reposait surtout sur une observation humaine, souvent subjective et chronophage. Pour surmonter cette limite, la reconnaissance et la détection automatiques des gestes chirurgicaux de base, appelés *surgèmes*, deviennent essentielles pour créer des systèmes d'évaluation objectifs, standardisés et automatisés.

Cependant, la classification automatique de ces actions reste un défi scientifique et technique majeur. Un geste chirurgical filmé ne se limite pas à une série d'images indépendantes ; il combine des caractéristiques visuelles complexes liées à l'espace et une dynamique qui évolue dans le temps. Pour relever ce défi spatio-temporel, les approches récentes en apprentissage profond privilégient des architectures capables de traiter simultanément ces deux dimensions, souvent en s'appuyant sur des bases de données de référence internationales.

Ce travail vise à explorer et à mettre en œuvre une solution technologique performante, basée sur une architecture hybride combinant les réseaux de neurones convolutifs (CNN) et les réseaux Long Short-Term Memory (LSTM), afin de classer efficacement les mouvements chirurgicaux.

Pour structurer notre démarche, ce mémoire s'organise en trois chapitres principaux, présentés comme suit :

- **Premier chapitre :État de l'art.**

Ce premier chapitre plante le décor en présentant le cadre général et en faisant le point sur les travaux existants. Il commence par introduire les concepts clés de la chirurgie et de l'apprentissage automatique, en s'attardant sur l'apprentissage supervisé et les défis liés à la classification. Il aborde ensuite l'importance de reconnaître les gestes chirurgicaux pour évaluer les compétences, en s'appuyant sur la base de référence JIGSAWS (JHU-ISI Gesture Skill Assessment Working Set). Enfin, il examine les méthodes actuelles de traitement des données cinématiques et vidéo, en détaillant des architectures comme les réseaux convolutifs ResNet18 , ainsi que les réseaux récurrents. Il analyse aussi comment la fusion de données multimodales et les mécanismes d'attention temporelle améliorent la reconnaissance.

- **Deuxième chapitre : fondements théoriques des modèles hybrides CNN-LSTM.**

Le deuxième chapitre établit les bases théoriques et mathématiques des technologies au cœur de notre solution hybride. Il débute par une explication du fonctionnement des réseaux de neurones convolutionnels (CNN), utilisés comme extracteurs automatiques de caractéristiques spatiales. Le rôle des couches de convolution, de pooling, des fonctions d'activation comme ReLU ou sigmoïde, ainsi que celui des couches entièrement connectées

pour la classification, y est clairement détaillé. Ensuite, le chapitre aborde les réseaux récurrents Long Short-Term Memory (LSTM), conçus pour capturer les dépendances temporelles à long terme dans des séquences de données, grâce à un mécanisme interne basé sur un état cellulaire et trois portes, contournant le problème de la disparition du gradient. Enfin, l'étude présente l'approche hybride CNN-LSTM, montrant comment la mise en cascade de ces deux architectures permet de modéliser de manière robuste des flux d'informations spatio-temporels complexes.

- **Troisième chapitre : expérimentations et résultats du modèle hybride CNN-LSTM sur la base de données JIGSAWS.**

Le troisième et dernier chapitre constitue la partie pratique et expérimentale de ce travail. Il présente d'abord la base de référence JIGSAWS, en analysant trois gestes chirurgicaux fondamentaux : la suture, le passage de l'aiguille et le nouage. Ensuite, il détaille le prétraitement des séquences temporelles, incluant normalisation, segmentation et fenêtrage avant leur introduction dans le modèle. Ce chapitre décrit également l'ensemble du protocole expérimental : architecture choisie, configuration des hyperparamètres (taille des filtres, taux d'apprentissage, optimiseurs), stratégie de validation et environnement matériel et logiciel (Python). Enfin, les performances sont présentées via des courbes d'apprentissage, matrices de confusion et métriques d'évaluation, permettant une comparaison des trois gestes avant une discussion critique sur les atouts et limites du modèle, notamment face au risque de sur-apprentissage.

En définitive, cette introduction jette les fondations d'une réflexion qui nous conduira des concepts théoriques jusqu'à leur mise en œuvre concrète. En structurant notre démarche autour d'un état de l'art solide, d'une analyse approfondie des réseaux CNN et LSTM, puis d'une validation expérimentale sur des données chirurgicales normalisées, ce mémoire entend prouver la pertinence et l'efficacité des architectures profondes hybrides face aux enjeux déterminants de la médecine de demain.

Chapitre 1

État de l'art

1.1 Introduction

L'intelligence artificielle et l'apprentissage automatique occupent aujourd'hui une place centrale dans de nombreux domaines scientifiques et médicaux. Ce chapitre vise à introduire les concepts fondamentaux nécessaires à la compréhension de notre étude, et présente les notions essentielles liées à la chirurgie, à l'intelligence artificielle et l'apprentissage automatique, en mettant l'accent sur l'apprentissage supervisé et la classification.

1.2 Chirurgie et gestes chirurgicaux

1.2.1 Définition

La reconnaissance des actions chirurgicales joue un rôle important dans la formation chirurgicale moderne [9, 10]. Plus précisément, la détection automatique des gestes chirurgicaux est d'une importance capitale pour l'évaluation des compétences et le développement de cadres de formation automatisés. Les avancées récentes se concentrent principalement sur les actions robotiques effectuées dans un environnement simulé, en utilisant la vidéo et/ou les données cinématiques des instruments[11], sur la base du célèbre ensemble de données JIGSAWS (JHU-ISI GESTURE SKILL ASSESSMENT WORKING SET)[1]. La figure ci-dessous représente des segments de mouvement très détaillés, réalisés dans un but chirurgical spécifique (par exemple : saisir l'aiguille, pousser l'aiguille à travers le tissu)fig1.1.

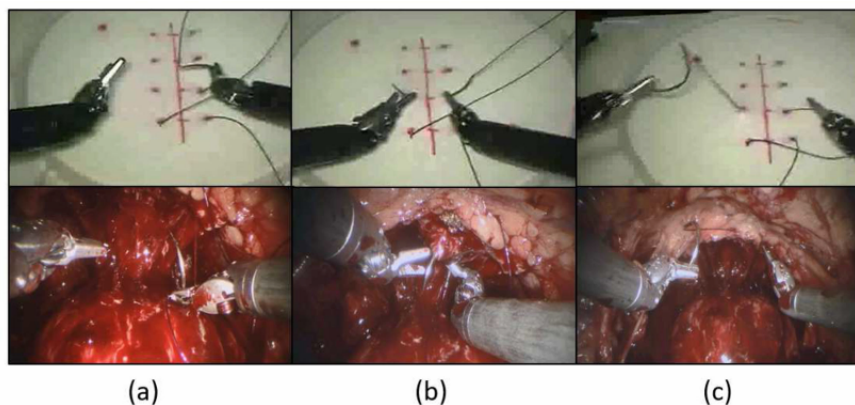


FIG. 1.1 : Exemples de surgèmes dans des environnements chirurgicaux réels et sur fantômes : (a) positionnement de la pointe de l'aiguille sur le point d'insertion, (b) passage de l'aiguille à travers le tissu, (c) retrait de l'aiguille hors du tissu [1].

1.2.2 Les différentes méthodes de reconnaissance des gestes

A. Reconnaissance des gestes chirurgicaux - Modèles temporels

La recherche actuelle se concentre sur des solutions plus puissantes basées sur l'apprentissage profond (Deep Learning), et en particulier sur les modèles temporels convolutionnels et récurrents. Ces derniers fonctionnent efficacement sur des données d'entrée de faible dimension, comme les trajectoires cinématiques ou les caractéristiques visuelles de haut niveau encodées par des réseaux de neurones convolutionnels 2D (CNN 2D) [12] ou 3D [13].

Les convolutions temporelles sont souvent utilisées dans des réseaux de type encodeur-décodeur, où les prédictions d'actions sont générées simultanément pour tous les horodatages. Le réseau TCN (Temporal Convolution Network) [14, 15] utilise une cascade de convolutions temporelles et de couches de regroupement/suréchantillonnage (pooling/upsampling) pour capturer les corrélations temporelles des données d'entrée à plusieurs niveaux hiérarchiques. Pour éviter la perte d'informations détaillées (fine-grained), des solutions à « deux flux » (two-stream) traitent les données à deux échelles temporelles différentes et fusionnent les informations à plusieurs niveaux de traitement [16, 17]. La résolution de ce problème repose sur une technique d'empilement de plusieurs couches de convolutions temporelles atrous (ou à trous) avec un facteur de dilatation croissant, augmentant ainsi le champ récepteur temporel du réseau sans avoir recours aux opérations de pooling [18, 19].

En revanche, les modèles récurrents tels que les LSTM (Long Short-Term Memory) [20] et les MS-RNN (Multi-Scale Recurrent Neural Network) [21] sont construits autour de cellules de mémoire capables de stocker des informations à long terme sur les observations passées. En dernier lieu, les approches hybrides intégrant des convolutions temporelles et des structures récurrentes qu'elles soient intégrées de manière séquentielle [22] ou en parallèle [23], ont montré d'excellentes capacités de reconnaissance.

B. Reconnaissance des gestes chirurgicaux - Fusion de capteurs

L'apprentissage conjoint et la fusion de données multimodales (vidéo, cinématique et flux optique) pour la reconnaissance des gestes ont été étudiés dans la littérature [24, 25]. Les résultats suggèrent que les modèles multimodaux offrent de meilleures performances que leurs équivalents unimodaux. La plupart des travaux connexes ont abordé la fusion de capteurs par une simple concaténation des caractéristiques unimodales [26, 27]. Cependant, cette méthode peut s'avérer sous-optimale en raison des différences de propriétés sémantiques et stochastiques entre les sources. Seules quelques études ont exploré des interactions plus complexes entre les différents flux de données. Le modèle Fusion-KV [23], par exemple, se compose de modèles de reconnaissance fonctionnant en parallèle sur différentes sources de données. La fusion de l'information s'effectue au moment du test (testing time), où les prédictions individuelles sont pondérées selon un système de vote basé sur la performance d'entraînement de chaque modalité par classe. Néanmoins, cette pondération par classe ne permet pas de capturer des interactions plus détaillées ni leur évolution dans le temps. L'intégration dynamique des représentations (embeddings) visuelles et cinématiques de haut niveau a été réalisée via un module d'apprentissage par graphes relationnels (MRG-Net) [28], visant à capturer des connaissances conjointes pour produire des représentations unimodales affinées. De manière similaire, l'attention multimodale est employée pour rechercher une coopération multimodale en temps opportun et affiner les représentations cachées afin d'obtenir une reconnaissance des gestes plus précise.

C. Apprentissage Multimodal Temporel Basé sur l'Attention

L'apprentissage multimodal temporel (TML)[29] vise à fusionner simultanément les informations provenant de plusieurs modalités et à modéliser la dynamique temporelle dans les données séquentielles. Les approches basées sur l'attention pour le TML ont été explorées pour la classification de vidéos[29, 30] et le sous-titrage automatique de vidéos (video captioning) à partir de signaux visuels, de mouvement et audio[31, 32]. Une solution simple mais efficace consiste à obtenir une représentation globale de chaque séquence d'entrée via des modèles temporels indépendants (par exemple, un LSTM) dotés de mécanismes d'attention, puis à même de fusionner ces représentations de haut niveau pour la classification vidéo [30]. L'avantage est que chaque modalité apprend de manière autonome à porter son attention sur différents segments temporels, tandis que les interactions multimodales sont tout de même capturées lors des étapes finales de traitement (late fusion). Des stratégies de fusion plus avancées étendent le mécanisme d'attention : elles ne se contentent plus de localiser les fenêtres temporelles pertinentes, mais pondèrent dynamiquement la contribution de chaque modalité de données au cours du temps [29, 31, 32]. Des poids dynamiques peuvent être attribués à chaque modalité en fonction de la correspondance entre l'entrée actuelle et la représentation multimodale précédente [29], ou avec tous les mots générés précédemment (pour le sous-titrage[31, 32]).

1.3 Intelligence artificielle

1.3.1 Définition

Le terme intelligence artificielle ou IA, est utilisé pour décrire l'émulation de l'intelligence humaine au sein de machines conçues pour penser et apprendre de manière similaire aux humains [33]. L'intelligence artificielle représente la capacité de machines ou de robots à effectuer des opérations nécessitant généralement l'intelligence humaine (telles que la reconnaissance vocale, la prise de décision, la perception visuelle et la compréhension du langage naturel).

1.3.2 Aspect aux techniques de l'intelligence artificielle

Aujourd'hui l'intelligence artificielle est considérée comme une branche de l'ingénierie qui met en œuvre des concepts et des solutions novateurs pour résoudre des défis complexes. Grâce aux progrès continus de la vitesse et de la capacité de l'électronique, ainsi que de la programmation logicielle, les ordinateurs pourraient un jour devenir aussi intelligents que les humains. la cybernétique d'aujourd'hui a grandement contribué au développement de l'IA [34]. À partir des définitions spécifique propres à chaque discipline, On peut représenter la structure de l'intelligence artificielle par :

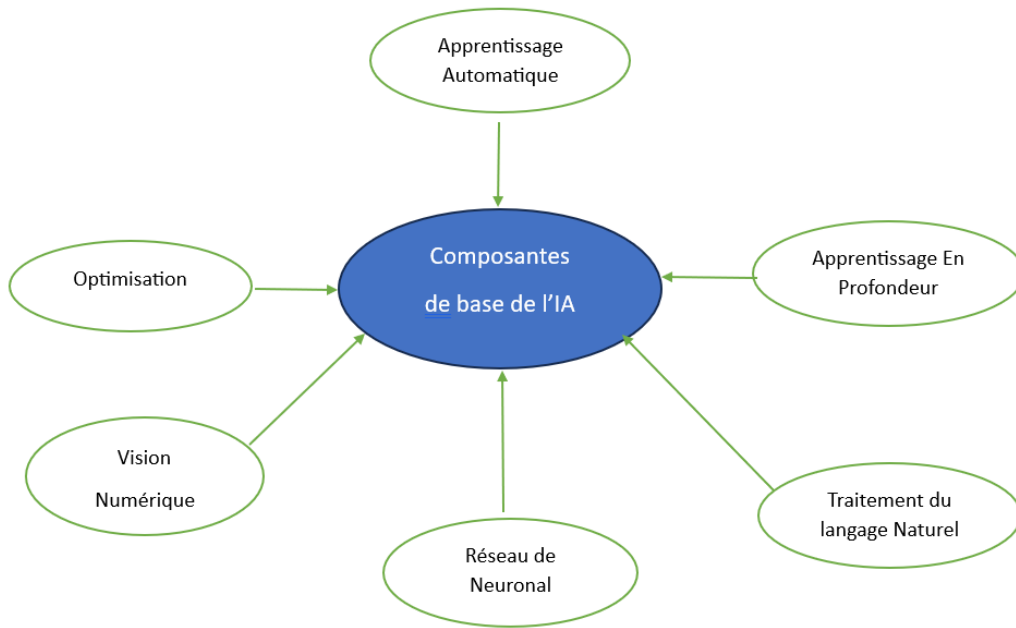


FIG. 1.2 : Les composantes de l'intelligence artificielle

1.4 Apprentissage automatique

1.4.1 Définition

L'apprentissage automatique (Machine Learning ou ML) est une branche de l'informatique et de l'intelligence artificielle qui utilise des méthodes statistiques pour permettre aux ordinateurs d'acquérir des connaissances et d'améliorer leurs performances sans nécessiter de programmation directe. Il repose sur le concept selon lequel les ordinateurs peuvent apprendre à partir de données, identifier des motifs (patterns) et tirer des conclusions avec une intervention humaine minimale[11, 35, 36].

1.4.2 Types

A. Apprentissage supervisé

L'apprentissage supervisé (Supervised Learning ou SL) est une approche de l'apprentissage automatique qui exploite des données étiquetées pour entraîner un système à prédire des résultats en fonction de son apprentissage. Il imite étroitement le processus d'apprentissage humain sous la direction d'un instructeur, en utilisant des exemples spécifiques pour en déduire des principes. L'apprentissage supervisé SL est généralement divisé en deux catégories principales généraux[36, 37, 38].

* Régression :

La régression est un terme utilisé en statistiques désignant un type d'analyse statistique qui vise à comprendre la relation entre une variable dépendante (variable de réponse) et une ou plusieurs variables indépendantes (prédicteurs), comme pour les tendances du marché ou les prévisions météorologiques. Le type le plus courant est la régression linéaire[38, 39].

* Classification :

La classification est une technique de SL qui consiste à catégoriser des données dans des classes distinctes. Il s'agit d'un processus récursif qui reconnaît et regroupe des objets de données dans des catégories ou des étiquettes prédéfinies[40]. Cette technique est utilisée pour prédire le résultat d'un problème donné en fonction des caractéristiques d'entrée (features). Elle peut être appliquée à des données structurées ou non structurées, et les classes sont communément appelées cibles (targets), étiquettes (labels) ou catégories. L'objectif de la classification est d'assigner un motif (pattern) inconnu à une classe connue. Par exemple, classer des e-mails comme « spam » ou « non-spam » est une application courante de la classification[41]. Les algorithmes de classification et de régression peuvent tous deux être utilisés pour la prédiction en apprentissage automatique et fonctionnent avec des ensembles de données étiquetés. Cependant, la distinction entre la classification et la régression réside dans la manière dont elles sont appliquées à des problèmes spécifiques d'apprentissage automatique.

B. Apprentissage non-supervisé

L'apprentissage non supervisé (Unsupervised Learning) consiste à entraîner la machine sans connaître le résultat attendu, en utilisant uniquement des échantillons d'entrée sans étiquettes. Il découvre des motifs (patterns) dans les données et crée ses propres regroupements de données (clusters). Il existe deux types principaux et plus courants d'algorithmes d'apprentissage non supervisé : le partitionnement (clustering) et l'analyse d'association. Cette technique est utile pour identifier des motifs inconnus dans les données, comme les moteurs de recommandation sur les boutiques en ligne qui s'appuient sur l'apprentissage non supervisé, plus précisément sur une technique appelée clustering [36, 42].

C. Apprentissage semi-supervisé

L'apprentissage semi-supervisé (Semi-Supervised Learning ou SSL) est une technique puissante qui combine l'apprentissage supervisé et non supervisé pour améliorer la précision de l'apprentissage. Il répond au défi du manque de données étiquetées en tirant parti de l'abondance de données non étiquetées [36, 42]. Le SSL trouve des applications dans divers domaines tels que l'analyse de la parole, la classification de contenu Web, la classification de séquences de protéines et les classificateurs de documents textuels [43].

D. Apprentissage par renforcement

L'apprentissage par renforcement (Reinforcement Learning ou RL) est une forme unique d'apprentissage automatique qui permet aux machines et aux agents logiciels de déterminer le comportement optimal afin d'obtenir les résultats souhaités [43]. Contrairement aux approches traditionnelles, le RL repose sur des expériences par essais et erreurs en recevant un retour d'information (feedback) qui renforce les résultats favorables et décourage les résultats défavorables. À mesure que l'agent accumule de l'expérience, il s'améliore et devient plus performant dans la prise de décisions pour atteindre son objectif [44]. Dans ce cadre, un agent interagit avec un environnement dynamique, recevant des récompenses et des punitions pour atteindre un but spécifique sans l'intervention explicite d'un enseignant [42]. Bien qu'il soit parfois considéré comme un sous-ensemble de l'apprentissage semi-supervisé, le RL est largement reconnu comme une forme distincte d'apprentissage automatique, étudiée à travers diverses disciplines en raison de sa généralité [45]. Ces disciplines incluent la théorie des jeux, la théorie du contrôle, la recherche opérationnelle, la théorie de l'information, l'optimisation par simulation, les systèmes multi-agents, l'intelligence en essaim, les statistiques et les algorithmes génétiques [36, 39].

1.4.3 Méthodes basées sur la classification des gestes chirurgicaux

A. Données vidéo en entrée

Andru P. Twinanda et al [46] ont proposé une méthode de reconnaissance des phases chirurgicales utilisant un réseau de neurones convolutifs (CNN) appelé EndoNet pour apprendre automatiquement des caractéristiques à partir de vidéos de cholécystectomie. Dans un travail similaire, Colin Lea et al [2] ont proposé une méthode de segmentation et de reconnaissance d'actions chirurgicales utilisant un CNN spatio-temporel. La figure 1.3 illustre le schéma de leur modèle, qui se compose de trois éléments : des composants spatiaux, segmentaux et temporels. Bien que les auteurs de ces deux contributions aient démontré l'efficacité de leurs méthodes pour la reconnaissance de phases et la segmentation d'actions fines, l'évaluation des compétences chirurgicales n'est pas abordée, alors que nous nous concentrons exclusivement sur cette dernière. Un autre travail traitant de la reconnaissance des phases chirurgicales a été réalisé par Manish Sahu et al [47] ont proposé un CNN pré-entraîné avec une méthode d'apprentissage par transfert pour la détection et l'estimation du flux de travail chirurgical. Dandan Zhang et al [48] ont développé un réseau de neurones récurrents (RNN) bidirectionnel multicouche indépendant pour la reconnaissance des gestes chirurgicaux, combiné à un modèle de réseau de neurones convolutifs profonds (DCNN) basé sur l'architecture VGG [49] (dont l'architecture est présentée en figure 1.4) pour l'extraction de caractéristiques spatiales à partir des images vidéo chirurgicales. Les auteurs ont utilisé l'ensemble de données JIGSAWS pour valider leur méthode qui couvre la reconnaissance des gestes chirurgicaux. Enfin, Lingling Tao et al [50] ont présenté six techniques basées sur une approche temporelle pour la segmentation et la reconnaissance de gestes en chirurgie robotique, incluant les modèles de Markov cachés (HMM) et les Semi-HMM. Cependant, Isabel Funke et al [51] ont proposé des réseaux de neurones convolutifs 3D pour classer des séquences (snippets — un lot de quelques images consécutives extraites d'une vidéo chirurgicale) issues du jeu de données JIGSAWS en trois classes d'expertise.

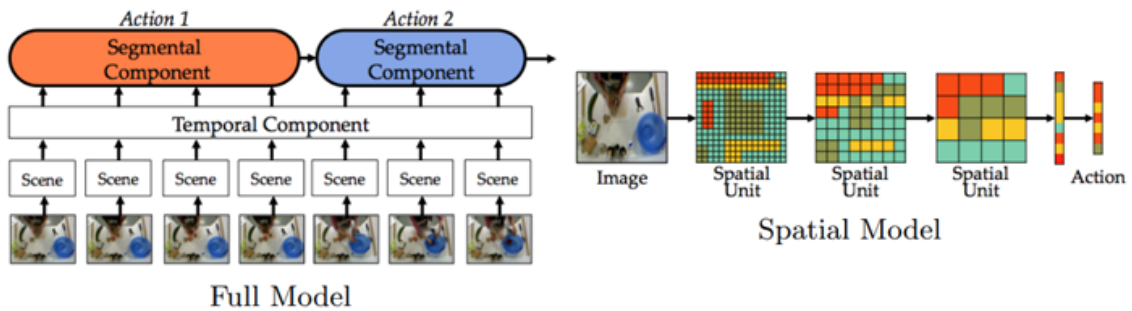


FIG. 1.3 : Architecture du modèle proposé avec ses trois composantes (spatiale, temporelle et segmentaire) et focus sur la composante spatiale[2].

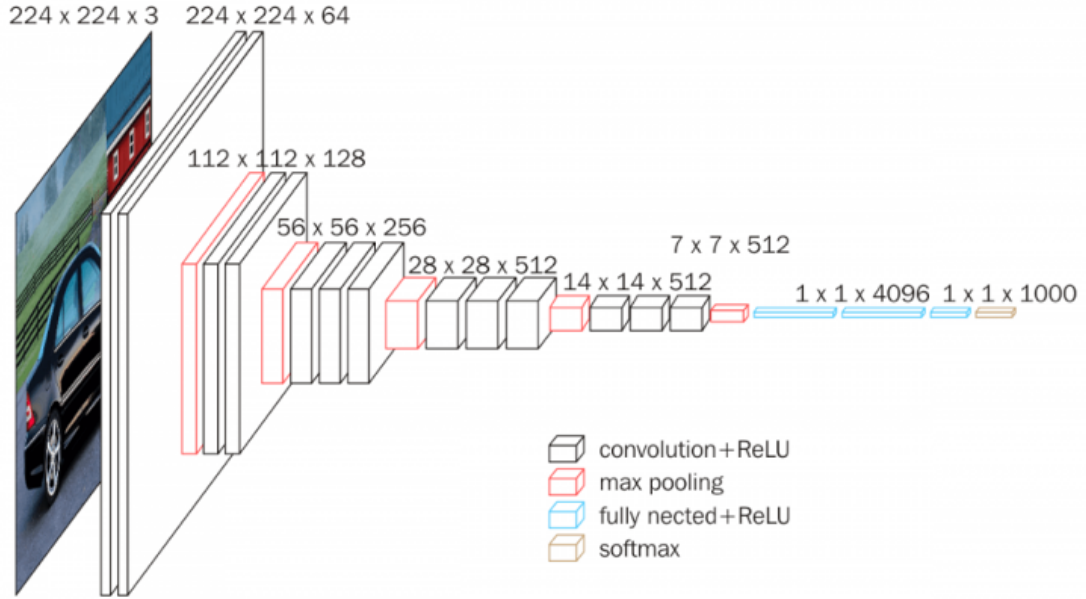


FIG. 1.4 : Architecture du réseau de neurones convolutif VGG[3].

B. Données cinématiques comme entrées

Robert Dipietro et al [52] ont proposé un travail sur la reconnaissance des activités chirurgicales à partir des données cinématiques du robot en utilisant des modèles LSTM et BiLSTM. Les mêmes auteurs principaux ont développé une architecture basée sur l'apprentissage non supervisé qui combine un encodeur-décodeur RNN et des réseaux de densité de mélange (MDN) pour modéliser la distribution conditionnelle du mouvement futur par rapport au mouvement passé. Ils ont ainsi démontré la possibilité d'apprendre des représentations significatives du mouvement lié aux compétences chirurgicales, sans supervision, en apprenant à prédire le futur. Dans un autre travail, Robert DiPietro et al [53] ont effectué une comparaison entre quatre architectures de RNN (les RNN simples, les Long Short-Term Memory, les Gated Recurrent Units et les RNN à historique mixte) dans le but de reconnaître des activités chirurgicales à partir de données cinématiques. Ilker Gurcan et al [21] ont soutenu que les RNN et les LSTM ne sont pas efficaces pour capturer la relation des caractéristiques à différentes échelles temporelles, ce qui conduit à des performances de reconnaissance sous-optimales pour les activités chirurgicales contenant des mouvements complexes à plusieurs échelles de temps. Ainsi, un réseau de neurones récurrents multi-échelles (MS-RNN), combinant la puissance des opérations de diffusion par ondelettes (wavelet scattering) et des réseaux LSTM, a été développé dans le but de reconnaître les activités chirurgicales.

1.5 Conclusion

Dans ce chapitre, nous avons présenté les notions fondamentales liées à la chirurgie et aux gestes chirurgicaux, ainsi que les bases de l'intelligence artificielle et de l'apprentissage automatique. On a également définie les principales approches d'apprentissage, ainsi que les différentes méthodes de classification utilisées dans ce domaine. Ce chapitre descriptif a permis de situer l'état de l'art relatif à notre thématique afin d'introduire plus précisément la démarche technique adoptée par la suite.

Chapitre 2

Fondements théoriques des modèles hybrides CNN-LSTM

Introduction

La classification des gestes chirurgicaux nécessite une analyse simultanée de la forme (aspect spatial) et du mouvement (aspect temporel). Ce chapitre présente les bases théoriques des modèles hybrides CNN-LSTM employés dans cette classification.

Nous introduisons les Réseaux de Neurones Convolutionnels (CNN), indispensables pour l'encodage spatial des images vidéo. Nous détaillerons ensuite les réseaux Long Short-Term Memory (LSTM) pour la modélisation des dépendances temporelles. En dernier lieu, nous présenterons l'architecture hybride CNN-LSTM, qui fusionne ces deux approches pour assurer une reconnaissance précise et fluide des gestes chirurgicaux.

2.1 Réseaux de Neurones Convolutifs

2.1.1 Définition

Les CNN sont appliqués à diverses tâches, notamment la détection d'objets, la reconnaissance vocale, la vision par ordinateur, la classification d'images et la bioinformatique [54]. Ils ont également démontré leur succès dans les tâches de prédiction de séries temporelles [55]. Les CNN sont des réseaux de neurones à propagation directe (feedforward) qui exploitent des structures convolutionnelles pour extraire des caractéristiques à partir des données [56]. Un CNN possède une architecture à deux étages qui combine un classificateur et un extracteur de caractéristiques, permettant une extraction automatique des caractéristiques et un entraînement de bout en bout avec le minimum de prétraitement nécessaire [57]. Contrairement aux méthodes traditionnelles, les CNN apprennent et reconnaissent automatiquement les caractéristiques des données sans avoir besoin d'une extraction manuelle par les humains [58]. La conception des CNN s'inspire de la perception visuelle [56]. Les principaux composants des CNN incluent la couche de convolution, la couche de pooling, la couche entièrement connectée et la fonction d'activation [59, 60]. La figure 2.1 présente le pipeline du réseau de neurones convolutionnel, montrant comment chaque couche contribue au traitement efficace et à la progression réussie des données d'entrée à travers le réseau.

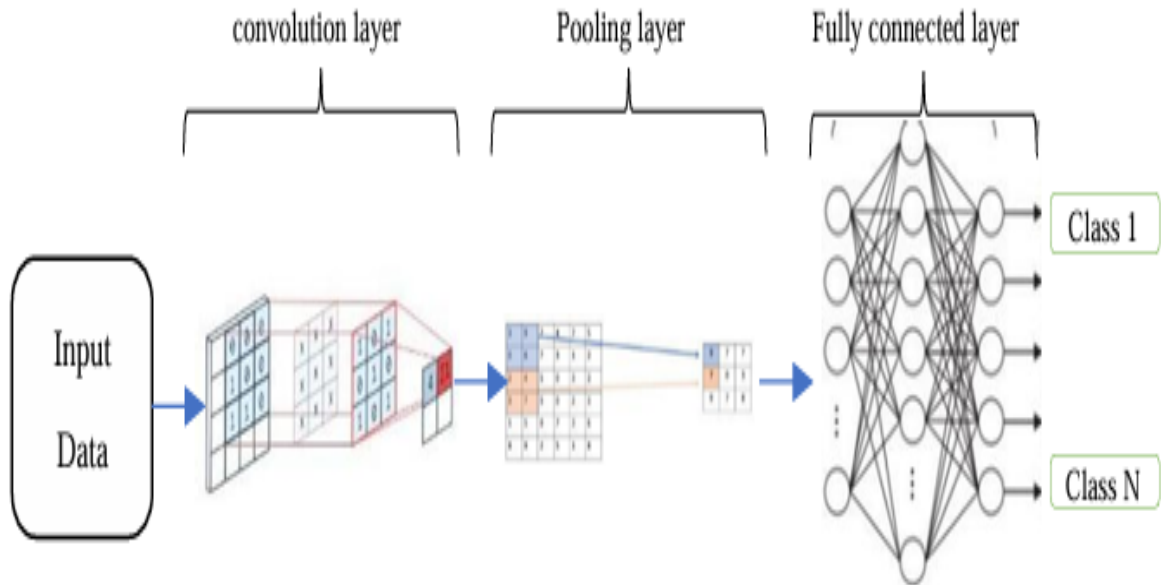


FIG. 2.1 : Le pipeline d'un réseau de neurones convolutionnel (CNN)[4]

2.1.2 Composants de base d'un CNN

A. Couche de convolution

La couche de convolution, élément clé des CNN, permet, via son empilement en multiples couches, d'extraire différentes caractéristiques des données d'entrée. En classification d'images, les couches inférieures ont tendance à capturer des caractéristiques de base telles que la texture, les lignes et les contours, tandis que les couches supérieures extraient des caractéristiques plus abstraites. La couche de convolution comprend des noyaux de convolution aptes à l'apprentissage, qui sont des matrices de poids généralement de longueur et largeur égales et d'un nombre impair (par exemple 3×3 , 5×5 ou 7×7). Ces noyaux sont convolués avec les cartes de caractéristiques d'entrée, glissant sur les régions de la carte de caractéristiques et exécutant des opérations de convolution [5]. La figure 2.2 illustre le schéma du processus de convolution.

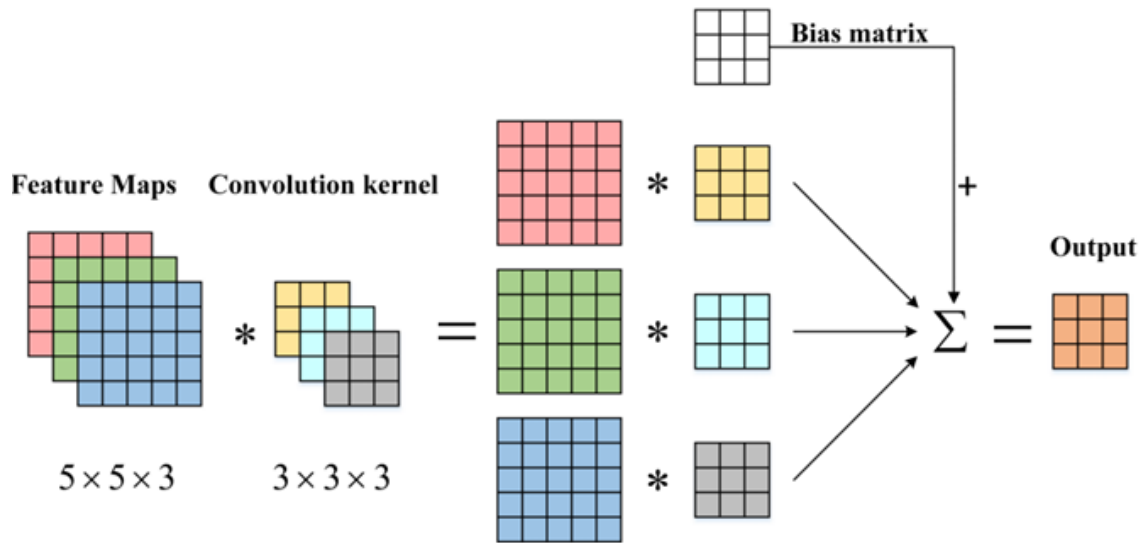


FIG. 2.2 : Schéma du processus convolution [5].

B. Couche de de pooling

Généralement située après la couche de convolution, la couche de pooling réduit le nombre de connexions dans le réseau en effectuant un sous-échantillonnage et une réduction de dimensionnalité sur les données d'entrée [61]. Son objectif principal est d'alléger la charge de calcul et de remédier aux problèmes de sur-apprentissage (overfitting) [62]. De plus, la couche de pooling permet aux CNN de reconnaître des objets même lorsque leurs formes sont déformées ou vues sous différents angles, en intégrant diverses dimensions d'une image grâce au pooling [63]. L'opération de pooling produit des cartes de caractéristiques de sortie plus robustes face aux distorsions et aux erreurs des neurones individuels [64]. Il existe différentes méthodes de pooling, notamment le Max Pooling, l'Average Pooling, le Spatial Pyramid Pooling, le Mixed Pooling, le Multi-Scale Order-Less et le Stochastic Pooling [65, 66]. La figure 2.3 illustre un exemple de Max Pooling, où une fenêtre glisse sur l'entrée, et le contenu de la fenêtre est traité par une fonction de pooling .

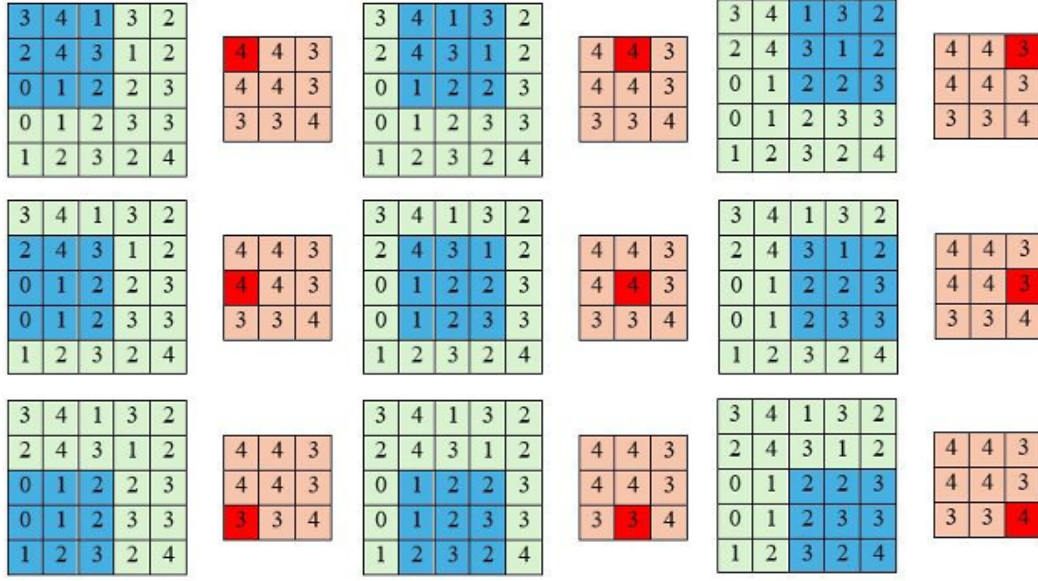


FIG. 2.3 : Calcul des valeurs de sortie d'une opération de max pooling 3×3 sur une entrée 5×5 [4].

C. Couche entièrement connectée

Dans cette couche, chaque neurone est connecté à tous les neurones de la couche précédente, conformément aux principes d'un réseau de neurones perceptron multicouche conventionnel. La couche FC reçoit son entrée de la dernière couche de pooling ou de convolution, qui est un vecteur créé par l'aplatissement des cartes de caractéristiques. La couche FC joue le rôle de classificateur, permettant ainsi au réseau de générer des prédictions. [67].

D. Fonctions d'activation

Les fonctions d'activation sont des composants fondamentaux des CNN, indispensables pour introduire la non-linéarité dans le réseau. Cette non-linéarité est cruciale pour la capacité du CNN à modéliser des motifs et des relations complexes au sein des données, lui permettant d'effectuer des tâches au-delà de la simple classification ou régression linéaire. En l'absence de fonctions d'activation non linéaires, un CNN ne pourrait effectuer que des opérations linéaires, restreignant ainsi fortement son aptitude à représenter avec précision la complexité des comportements non linéaires caractéristiques de nombreux phénomènes réels.[68].

La figure 2.4 illustre généralement comment ces fonctions d'activation modulent les signaux d'entrée pour produire une sortie, mettant en évidence les transformations non linéaires appliquées aux données d'entrée sur différentes régions de la courbe de la fonction. Dans cette figure, x_i représente la caractéristique d'entrée, tandis que w_{ij} désigne le poids associé à la connexion entre la caractéristique d'entrée x_i et le neurone j . La figure montre que le neurone j reçoit n caractéristiques simultanément. La sortie du neurone j est notée y_j , et son état interne, ou biais, est indiqué par b_j . La fonction d'activation, représentée par $f(\cdot)$, peut être de différents types tels que l'unité linéaire rectifiée (ReLU), la tangente hyperbolique (Tanh), la fonction sigmoïde, ou d'autres.

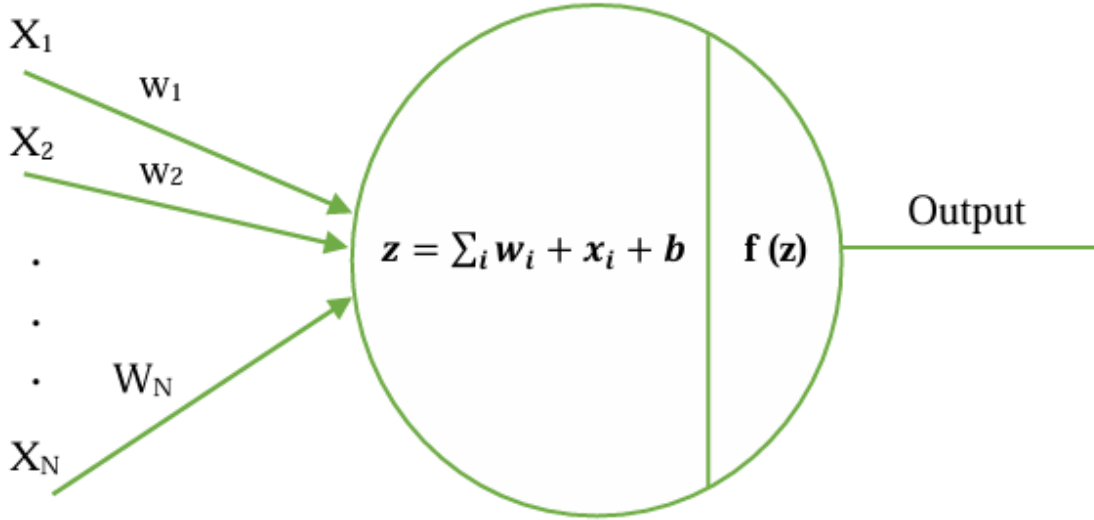


FIG. 2.4 : La structure générale des fonctions d'activation[4].

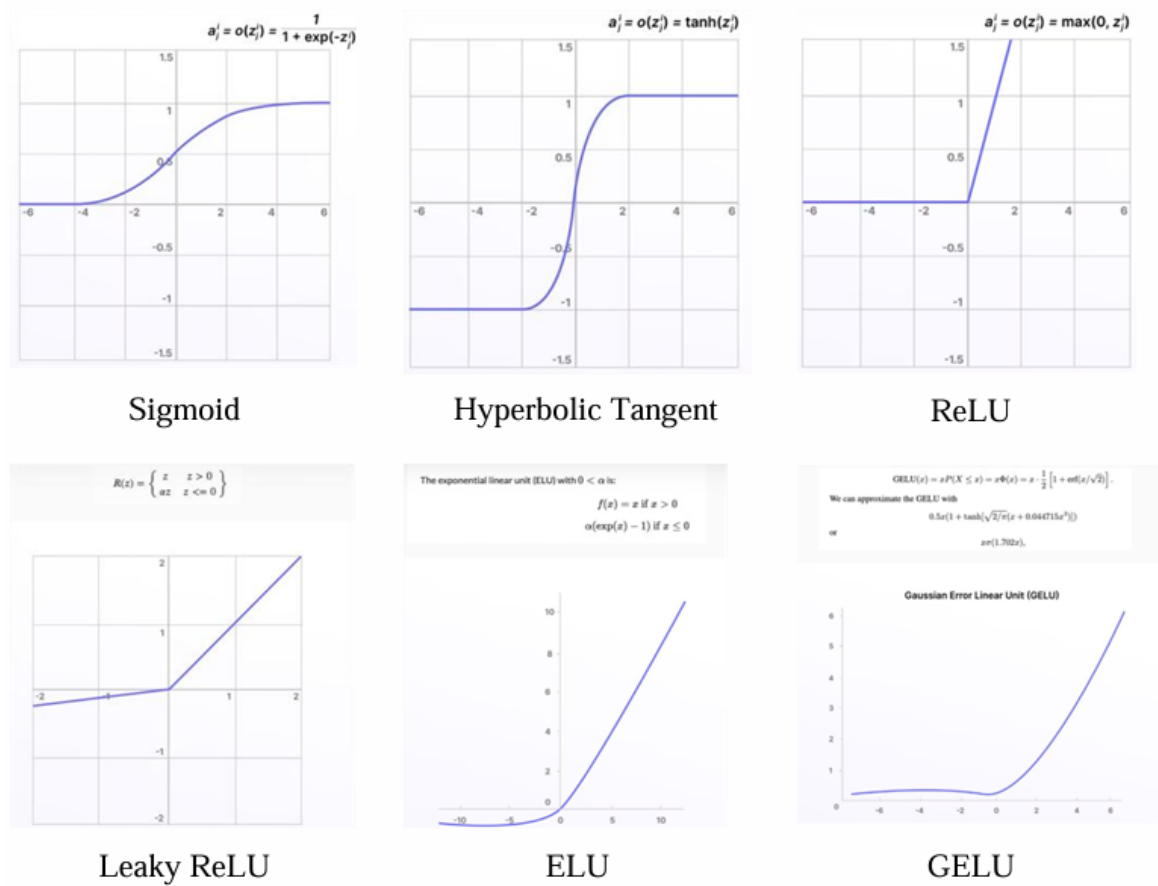


FIG. 2.5 : Diagramme des différentes fonctions d'activation [4].

Le ReLU (Rectified Linear Unit) est l'une des fonctions d'activation les plus utilisées dans les CNN modernes en raison de sa capacité à résoudre le problème de la disparition du gradient pendant l'entraînement. La définition mathématique du ReLU est donnée par eq.2.1,

où l'entrée du neurone est représentée par x [69] :

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{si } x_i \geq 0 \\ 0, & \text{si } x_i < 0 \end{cases} \quad (2.1)$$

Cette caractéristique aide les CNN à apprendre plus efficacement des représentations complexes en annulant toutes les valeurs d'entrée négatives tout en préservant les valeurs positives. Cela empêche également les neurones d'être saturés pendant l'entraînement.

En alternative, la définition de la fonction sigmoïde est représentée par eq.2.2, où x représente l'entrée du neurone :

$$f(x) = \frac{1}{e^{-x}} \quad (2.2)$$

Bien que la forme distinctive en S de la sigmoïde et sa capacité à condenser les nombres réels dans un intervalle compris entre 0 et 1 la rendent utile pour la classification binaire, sa tendance à saturer peut entraver l'entraînement en provoquant Le problème du gradient disparaissant dans les réseaux de neurones profonds.

2.1.3 Différentes Architecture de CNN

TAB. 2.1 : Comparaison entre différentes architectures de CNN populaires [8]

Architecture	Couches	Contribution principale	Points marquants	Avantages	Lacunes
LeNet-5 [1998]	7 (5 Conv + 2 FC)	Première architecture CNN populaire	Rapide et efficace pour les images à petite échelle	Corrélation spatiale pour réduire calculs et paramètres. Découverte automatique des hiérarchies.	Mise à l'échelle insuffisante ; filtres trop grands ; faible extraction.
AlexNet [2012]	8 (5 Conv + 3 FC)	Plus de profondeur et largeur que LeNet ; utilise ReLU, Dropout, pooling chevauchement ; GPUs	Comparable à LeNet-5 mais plus complexe, plus de filtres par couche	Extraction bas, moyen et haut niveau. Régularisation. Utilisation parallèle GPUs.	Neurones dormants ; artefacts de repliement.
ZFNet [2014]	8	Conceptualisation des niveaux intermédiaires	Utilisation de petits filtres 3×3 pour améliorer la précision	Visualisation des couches intermédiaires. Réduction taille/pas des premiers filtres.	Nécessite traitement supplémentaire pour visualisation.

Architecture	Couches	Contribution principale	Points marquants	Avantages	Lacunes
VGG [2014]	16–19 (13–16 Conv + 3 FC)	Structure homogène, petite taille de noyau ; profondeur accrue	Utilisation de petits filtres 3×3 dans chaque couche	Concept de champ réceptif effectif. Topologie simple et homogène.	Coût calculatoire élevé des couches FC.
GoogLeNet [2015]	22 Conv, 9 modules Inception	Concept de bloc ; séparation transformation/ fusion	Architecture plus profonde et large avec différentes tailles de champ réceptif	Filtres multi-échelles. Diviser/ transformer/ fusionner. Réduction paramètres. Classificateurs auxiliaires.	Réglage difficile. Risque de perte d'info par goulot.
Inception-V3 [2015]	42 Conv, 10 modules Inception	Résout le goulot ; remplace grands filtres par petits	Améliore l'efficacité ; Batch Normalisation accélère l'entraînement	Filtres asymétriques et couches bottleneck pour réduire coût.	Complexité architecturale ; absence d'uniformité.
ResNet [2016]	50, 101, 152	Connexions sautées ; mappage identitaire ; rétention long terme	Design unique avec skip connections et batch norm étendue	Réduction erreur réseaux profonds. Apprentissage résiduel. Atténue vanishing gradient.	Structure complexe ; dégradation info ; sur-adaptation hyperparamètres.
DenseNet[2017]	117 Conv, 3 transition, 1 classification	Transmission d'information entre couches ; blocs interconnectés	Toutes les couches connectées feed-forward ; atténue vanishing gradient	Flux max de données. Évite réapprentissage cartes redondantes. Caractéristiques bas/haut niveau disponibles.	Augmentation significative paramètres avec nombre de cartes.

2.2 Long Short-Term Memory (LSTM)

2.2.1 Définition

Les réseaux LSTM sont un type spécialisé de RNN conçu pour apprendre efficacement les dépendances à long terme dans les données séquentielles [70]. Les RNN traditionnels souffrent souvent du problème fuite de gradient (vanishin), ce qui limite leur capacité à capturer les informations provenant d'éléments éloignés dans une séquence. Les LSTM relèvent ce défi grâce à leur architecture unique et à leurs mécanismes de portes. Cette section fournit un aperçu complet de l'architecture LSTM, des mécanismes qui facilitent son fonctionnement, ainsi qu'une comparaison avec les RNN traditionnels [71].

2.2.2 Architecture et le mécanisme du LSTM

L'architecture des réseaux LSTM repose sur une structure élaborée dont l'objectif est de maîtriser et de préserver l'information tout au long de séquences temporelles étendues. Cette performance remarquable s'explique par la présence de plusieurs composants essentiels au sein de l'architecture :

A. État cellulaire (Cell State)

L'état cellulaire, noté C_t , constitue l'unité mémoire du LSTM. Il transporte les informations pertinentes tout au long de la séquence et est mis à jour à chaque pas de temps, il permet au LSTM de conserver l'information sur de longues périodes. L'état cellulaire est essentiel pour maintenir le contexte, car il peut stocker des informations issues des pas de temps précédents sans dégradation significative. Cette caractéristique permet aux LSTM de mémoriser les détails importants sur de longues séquences, ce qui les rend adaptés à des applications telles que la modélisation du langage et la prédiction de séries temporelles.

B. État caché (Hidden State)

L'état caché, noté h_t , correspond à la sortie du LSTM au pas de temps t et est utilisé pour effectuer des prédictions. Il encapsule l'information relative à la séquence d'entrée jusqu'à l'instant présent et est transmis aux cellules LSTM suivantes. L'état caché peut être interprété comme une version filtrée de l'état cellulaire, représentant l'information pertinente nécessaire à la prédiction courante. Cette nature dynamique de l'état caché permet au modèle d'adapter ses sorties en fonction du contexte évolutif de la séquence d'entrée.

C. Portes (Gates)

Les LSTM utilisent trois types de portes pour contrôler le flux d'information, chacune remplissant une fonction distincte :

D. Porte d'entrée (Input Gate - i_t) :

Cette porte détermine quelle quantité des nouvelles informations provenant de l'entrée courante doit être ajoutée à l'état cellulaire. Classiquement, La porte d'entrée utilise une fonction d'activation sigmoïde pour produire des valeurs comprises entre 0 et 1, agissant ainsi efficacement comme un filtre. La formule de la porte d'entrée est :

$$i_t = \sigma(W_i^T \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

Une valeur obtenue par la formule eq. ?? proche de un indique que l'information doit être entièrement ajoutée à l'état cellulaire, tandis qu'une valeur proche de zéro implique qu'il faut ajouter peu ou pas d'information. Ici, i_t représente l'activation de la porte d'entrée, W_i la matrice des poids, h_{t-1} l'état caché du pas de temps précédent, x_t l'entrée courante, et b_i le terme de biais.

E. Porte d'oubli (Forget Gate - f_t) :

La porte d'oubli décide quelles informations de l'état cellulaire doivent être écartées. Comme la porte d'entrée, elle emploie une fonction sigmoïde :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

La porte d'oubli obtenue par la formule eq.2.4 permet au LSTM de supprimer les informations non pertinentes, contribuant ainsi à éviter que l'état cellulaire ne s'encombre. Ce mécanisme est essentiel pour maintenir la capacité du modèle à se concentrer sur les motifs pertinents au fil du temps. Dans cette formule, f_t est l'activation de la porte d'oubli, W_f la matrice des poids, b_f le terme de biais, et les autres variables sont telles que définies précédemment.

F. Porte de sortie (Output Gate - o_t) :

Cette porte contrôle quelle partie de l'état cellulaire sera produite en tant qu'état caché. Elle détermine l'information transmise à la cellule LSTM suivante :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

La porte de sortie est, calculée par l'application de la formule eq.2.5, garantit que l'état caché ne reflète que les informations les plus pertinentes issues de l'état cellulaire, ce qui est crucial pour effectuer des prédictions précises à chaque pas de temps. Ici, o_t est l'activation de la porte de sortie, W_o la matrice des poids, b_o le terme de biais.

Les mises à jour de l'état cellulaire et de l'état caché sont régies par les équations suivantes :

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.6)$$

Dans la formule eq.2.6, $\tilde{C}_t$ représente les valeurs candidates pour l'état cellulaire, W_C est la matrice des poids, et b_C est le terme de biais.

$$C_t = f_t \otimes C_{t-1} + i_t \quad (2.7)$$

Dans l'équation eq.(2.7), C_t est l'état cellulaire mis à jour, C_{t-1} l'état cellulaire précédent, et $\otimes$ désigne la multiplication élément par élément.

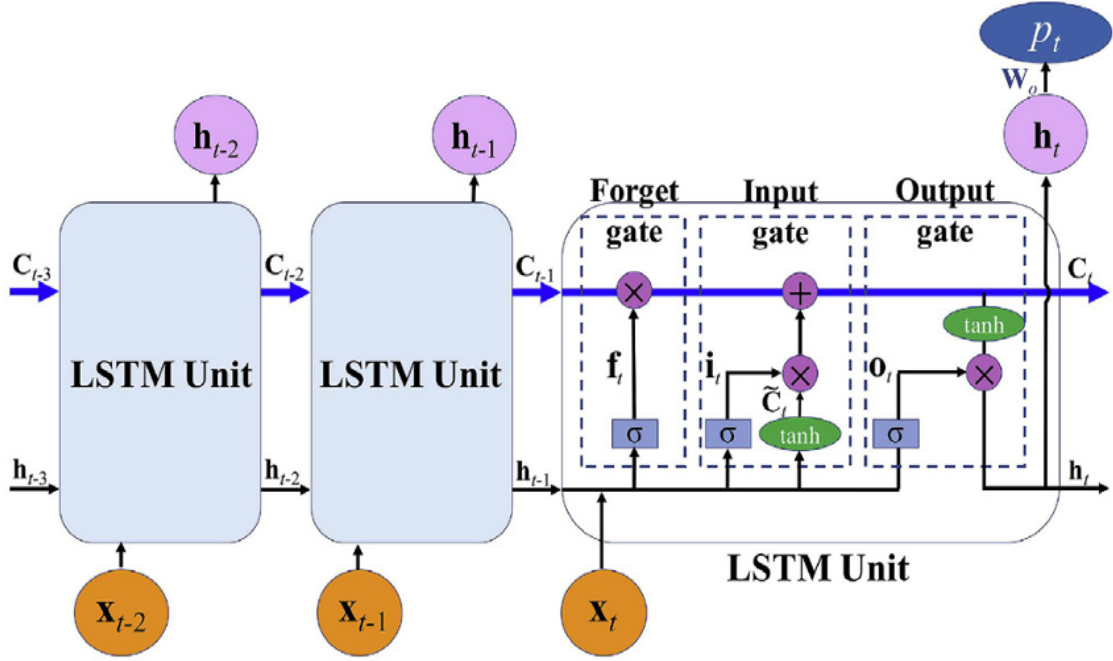


FIG. 2.6 : L'architecture de la cellule LSTM[6]

La figure 2.6 montre l'architecture d'une cellule LSTM. On y distingue :

- **Les entrées** : X_{t-2} , X_{t-1} et X_t (les données qui arrivent à différents moments)
- **La mémoire à court terme** : h_{t-2} , h_{t-1} et h_t (l'état caché)
- **La mémoire à long terme** : C_{t-3} , C_{t-2} , C_{t-1} et C_t (l'état de la cellule)
- **Trois portes** :
 - f_t : porte d'oubli (quoi effacer)
 - i_t : porte d'entrée (quoi ajouter)
 - o_t : porte de sortie (quoi transmettre)
- **Fonctions d'activation** : σ (sigmoïde) et $\tanh$ (tangente hyperbolique)
- **Poids** : W_o (matrice de poids pour la porte de sortie)
- **Vecteur d'activation** : p_t

2.3 Modèle hybride CNN-LSTM

2.3.1 Définition

L'architecture CNN-LSTM combine des couches de CNN, utilisées pour l'extraction de caractéristiques à partir des données d'entrée, avec des réseaux de neurones récurrents de type LSTM, permettant la modélisation des dépendances temporelles et la prédiction de séquences[72].

2.3.2 Architecture Hybride CNN-LSTM

Initialement, les modèles CNN-LSTM ont été développés pour des problèmes de prédiction sur des séries temporelles visuelles ainsi que pour la génération de descriptions textuelles à partir de séquences d'images, telles que les vidéos. Ces applications incluent notamment : (i) la reconnaissance d'activités, qui consiste à générer une description textuelle d'une action observée dans une séquence d'images ; (ii) la description d'images, visant à produire une description textuelle d'une image unique ; et (iii) la description de vidéos, qui consiste à générer une description textuelle d'une séquence d'images [72].

Les modèles CNN-LSTM constituent ainsi une classe d'architectures à la fois profondes sur le plan spatial et temporel, offrant une grande flexibilité pour diverses tâches de vision impliquant des entrées et sorties séquentielles.

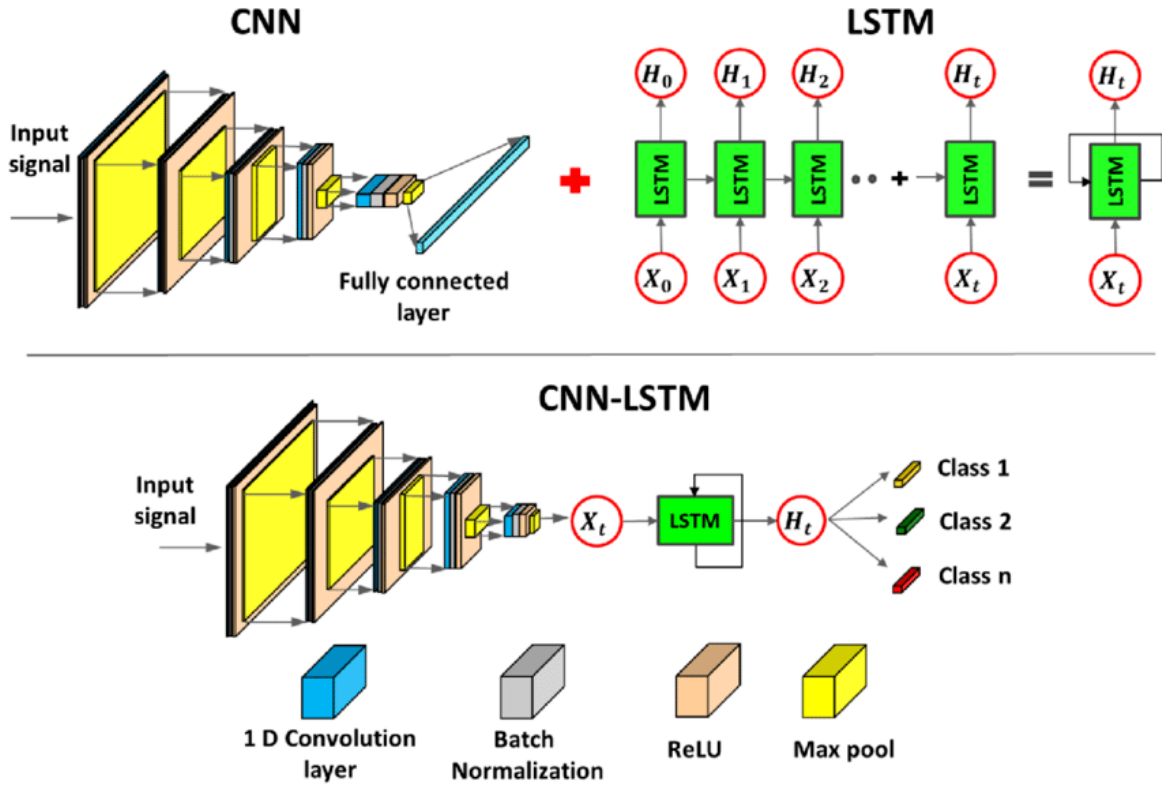


FIG. 2.7 : Architecture hybride CNN-LSTM[7]

2.3.3 Évaluation du modèle

Afin d'évaluer les performances d'un modèle d'estimation de la durée de vie restante (RUL) sur des données de test, l'erreur quadratique moyenne (RMSE), donnée par eq.2.8, est utilisée [73].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - x_0)^2} \quad (2.8)$$

Le coefficient de détermination R^2 , présenté dans eq.2.9, est également largement utilisé comme indicateur d'évaluation pour l'estimation de la RUL[73].

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (2.9)$$

où SS_{res} représente la somme des carrés des résidus, et SS_{tot} désigne la somme totale des carrés.

Une valeur de R^2 égale à 1 indique que le modèle explique parfaitement toute la variabilité de la variable cible, tandis qu'une valeur de 0 signifie que le modèle n'explique aucune variabilité. Dans la littérature, une fonction de score est également utilisée pour évaluer la qualité des modèles [74, 75] :

$$S = \begin{cases} \sum_{i=1}^n \left(e^{-\frac{h_i}{13}} - 1 \right), & \text{si } h_i < 0 \\ \sum_{i=1}^n \left(e^{\frac{h_i}{10}} - 1 \right), & \text{si } h_i \geq 0 \end{cases} \quad (2.10)$$

2.3.4 Mécanisme d'intégration hybride de base

A. Dans un pipeline CNN-LSTM standard :

- Préparation de l'entrée

Les données peuvent être des séries temporelles multivariées (forme : lot $\times$ temps $\times$ caractéristiques) ou des séquences d'images/vidéos.

- Extraction de caractéristiques CNN

- Les CNN 1D (pour les séries temporelles/données de capteurs) ou les CNN 2D/3D (pour les images/vidéos) appliquent des filtres pour détecter des motifs locaux.
- Plusieurs couches Conv + ReLU + Pooling construisent des représentations hiérarchiques.
- Les opérations de restructuration garantissent que la sortie maintient une structure séquentielle adaptée au LSTM [72].

- Transition vers LSTM

Les cartes de caractéristiques spatialement enrichies sont traitées comme une nouvelle séquence.

- Modélisation temporelle LSTM

- Traite la séquence des caractéristiques CNN à travers le temps.
- Utilise des cellules de mémoire et trois portes (oubli, entrée, sortie) pour retenir ou rejeter sélectivement l'information sur de longs horizons.
- Peut être empilé, bidirectionnel ou combiné avec des mécanismes d'attention pour améliorer les performances [73].

- Étape de sortie

Des couches entièrement connectées (Dense Layers) sont utilisées pour la classification ou la régression.

Cette intégration séquentielle (CNN $\rightarrow$ LSTM) est la caractéristique déterminante de l'hybride : le LSTM ne fonctionne pas directement sur des données brutes mais sur des caractéristiques de haut niveau déjà extraites par le CNN, ce qui améliore la généralisation et l'efficacité du modèle [76].

2.4 Conclusion

Ce chapitre a permis d'établir les bases théoriques nécessaires à la compréhension des modèles profonds utilisés dans ce travail. Nous avons d'abord détaillé les CNN , en explorant leur définition, leurs composants fondamentaux et les diverses architectures existantes pour l'extraction de caractéristiques spatiales.

Parallèlement, l'étude s'est portée sur les LSTM, dont nous avons analysé la définition, l'architecture spécifique et le mécanisme de fonctionnement interne permettant de gérer les séquences temporelles.

La synthèse de ces deux approches au sein de l'architecture hybride CNN-LSTM constitue le cœur de notre solution pour la classification des gestes chirurgicaux. En combinant l'analyse visuelle et la mémoire séquentielle. Cette maîtrise théorique nous permet désormais d'aborder la phase expérimentale, qui débutera au chapitre suivant par la présentation du jeu de données JIGSAWS.

Chapitre 3

Expérimentations et résultats du modèle hybride CNN-LSTM sur la base de données JIGSAWS

introduction

Dans ce chapitre, nous présentons l'expérience réalisées afin d'évaluer les performances du modèle hybride CNN-LSTM appliqué à la reconnaissance des gestes chirurgicaux à partir de la base de données JIGSAWS.

Nous commençons par une présentation générale de la base de données utilisée ainsi que des différents gestes chirurgicaux étudiés. Nous décrivons ensuite les étapes de prétraitement des données et le protocole expérimental adopté, incluant l'architecture du modèle, les hyperparamètres choisis et l'environnement logiciel et matériel utilisé.

Par la suite, nous exposons les résultats expérimentaux obtenus à travers plusieurs métriques d'évaluation telles que l'accuracy, la précision, le rappel et le F1-score. En dernier lieu, une discussion est menée afin d'interpréter les performances du modèle, d'identifier ses points forts ainsi que ses limites dans le contexte de la reconnaissance automatique des gestes chirurgicaux.

3.1 Présentation de la base de données JIGSAWS

3.1.1 Description générale de JIGSAWS

Le JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) est un jeu de données développé par des chercheurs de l'Université Johns Hopkins et de l'Information Sciences Institute [1]. Il rassemble des données cinétiques issues de gestes chirurgicaux réalisés avec le système robotique Da Vinci (DVSS) [77, 3], par des chirurgiens de trois niveaux d'expertise : experts, intermédiaires et novices. Les tâches couvertes incluent la suture, le passage d'aiguille et le nouage de nœuds.

Principalement destiné à la recherche sur l'évaluation des compétences, la formation chirurgicale et l'amélioration des systèmes robotiques, JIGSAWS met à disposition à la fois des données de mouvement des instruments robotiques et des enregistrements vidéo des interventions. Ces données permettent d'analyser les trajectoires des instruments, les mouvements des mains du chirurgien ainsi que la dynamique du site opératoire. Des métadonnées viennent compléter l'ensemble : type de procédure, instrument utilisé, phase de la tâche, etc.

Accessible publiquement, ce dataset est largement exploité dans la communauté de la robotique chirurgicale. Dans les sections qui suivent, nous détaillons le système Da Vinci ainsi que le contenu du dataset JIGSAWS.

3.1.2 Les trois gestes chirurgicaux étudiés

* Suturing(SU)

L'opérateur saisit l'aiguille, se dirige vers l'incision (représentée par une ligne verticale sur le modèle d'établi), et fait passer l'aiguille à travers le "tissu", en entrant par le point marqué d'un côté de l'incision et en ressortant par le point correspondant marqué de l'autre côté. Après le premier passage de l'aiguille, il l'extrait du tissu, la passe à la main droite et répète l'opération trois fois de plus.

* Needle-Passing(NP)

L'opérateur saisit l'aiguille (dans certains cas, ce geste n'est pas capturé dans la vidéo) et la fait passer à travers quatre petits anneaux métalliques, de la droite vers la gauche. Les anneaux sont fixés à une faible hauteur au-dessus de la surface du modèle d'établi.

* Knot-Tying (KT)

L'opérateur saisit l'une des extrémités d'un fil de suture attaché à un tube flexible (lui-même fixé par ses extrémités à la surface du modèle d'établi), et réalise un nœud à boucle simple.

3.1.3 Prétraitement des données

Afin d'assurer une bonne performance du modèle CNN-LSTM, plusieurs étapes de prétraitement des données sont nécessaires : la normalisation, la segmentation, le fenêtrage, et la préparation des séquences temporelles. Ces étapes permettent de structurer les données brutes en entrées adaptées à l'architecture du modèle.

Pour limiter le risque de sur-apprentissage, nous avons converti les images en niveaux de gris. Ce choix permet de neutraliser les variations de luminosité et les reflets chromatiques qui ne sont pas pertinents pour la classification. En réduisant la complexité des données, nous forçons le modèle à se concentrer exclusivement sur les formes et les textures des instruments, renforçant ainsi sa capacité de généralisation.

3.2 Protocole expérimental et paramétrage du modèle

3.2.1 Architecture du modèle CNN-LSTM retenue

A. CNN extracteur (ResNet18 pré-entraîné et gelé[78]).

- Suppression de la dernière couche fc (remplacée par Identity)
- Sortie : features de taille **512**

B. LSTM (1 couche, hidden_size = 128)

- input_size = 512 (features ResNet)
- hidden_size = 128

C. Classificateur final

- Linear(128 → 64) + ReLU + Dropout(0.3) + Linear(64 → num_classes)

3.2.2 Configuration des hyperparamètres

Afin de garantir la reproductibilité de nos résultats et d'optimiser l'apprentissage du modèle CNN-LSTM sur la base de données JIGSAWS, nous avons défini une configuration expérimentale rigoureuse.

A. Hyperparamètres du Modèle

Le tableau ci-dessous détaille la structure interne des deux composantes de notre réseau hybride :

TAB. 3.1 : la structure interne de réseau hybride CNN-LSTM

Composant	Paramètre	Valeur
Extracteur Spatial (CNN)	Taille des filtres	7×7 puis 3×3
	Nombre de filtres	64, 128, 256, 512
	État des poids	Gelés (Frozen)
Analyseur Temporel (LSTM)	Unités cachées (Hidden Size)	128
	Nombre de couches	1
	Directionnalité	Unidirectionnel
	Régularisation (Dropout)	0.5 (50%)

B. Paramètres d'Entraînement

L'optimisation a été réalisée avec une stratégie de Transfer Learning, où seul le bloc récurrent et la couche de classification sont mis à jour :

- **Optimiseur** : AdamW (utilisant un `weight_decay` de 10^{-4} pour la régularisation des poids).
- **Taux d'apprentissage (Learning Rate)** : 1×10^{-5} . Cette valeur faible permet une convergence stable sans diverger des caractéristiques extraites par le CNN.
- **Taille du batch** : 16 séquences par itération.
- **Nombre d'époques** : 20.
- **Fonction de perte** : Entropie croisée (Cross-Entropy Loss), adaptée pour la classification des 3 tâches de JIGSAWS.

C. Prétraitement et Fenêtrage Temporel

La dynamique des gestes chirurgicales impose un découpage spécifique des vidéos :

Résolution d'entrée : 224×224 pixels.

fréquence d'échantillonnage : 10 FPS (Frames Per Second).

Longueur de séquence (T) : 16 frames.

Méthode de segmentation : Fenêtre glissante (Sliding Window) avec un pas de 8 frames, générant un recouvrement temporel de 50% pour augmenter la densité des données d'entraînement.

3.2.3 Stratégie de validation

A. Répartition des Données (Split)

Pour garantir la robustesse du modèle et éviter toute fuite de données (data leakage), le jeu de données JIGSAWS a été segmenté en trois sous-ensembles distincts, selon les proportions définies dans notre configuration (config.py) :

* **Test Set (20%)** : `test_split = 0.2`. isole initialement 20% de l'ensemble des données globales pour constituer le jeu de test. Ces données sont strictement réservées à l'évaluation finale et restent totalement invisibles pour le modèle durant toute la phase d'apprentissage.

* **Validation Set (5%)** : Les 80% restants forment la base d'apprentissage. C'est sur cette base qu'est appliqué le paramètre `val_split = 0.05`. Ainsi, 5% de ces données d'entraînement sont prélevées pour former le jeu de validation.

* **Train Set (80%)** : L'ensemble d'entraînement constitue la base de données principale de notre projet, représentant 80% du jeu de données global initial.

B. Méthode d'évaluation de la performance

La phase d'évaluation finale, repose sur une approche standardisée en apprentissage profond :

C. Inférence réaliste (Évaluation stricte)

L'évaluation du modèle sur l'ensemble de test (Test Set) permet de réaliser une inférence réaliste. Ce protocole constitue une évaluation stricte, car le modèle est testé sur des vidéos totalement inédites, exclues de la phase d'apprentissage.

- **Capacité de généralisation** : Le modèle atteint une exactitude globale (Accuracy) de **95,00 %**. Ce score élevé prouve sa capacité à fonctionner efficacement en conditions réelles face à des données inconnues.

- **Analyse des erreurs** : En situation réelle, l'erreur du modèle reste mineure et localisée. La matrice montre que seulement **2 vidéos** de la classe Knot Tying ont été confondues avec le Suturing, tandis que le Needle Passing est détecté à **100 %**.

D. Métriques de performance retenues

Pour quantifier l'efficacité de notre modèle CNN-LSTM sur les 3 tâches chirurgicales principales, quatre métriques clés issues de la bibliothèque `scikit-learn (classification_report)` sont calculées :

- **Précision (Precision)** Proportion de gestes correctement prédits parmi l'ensemble des gestes prédits pour une classe donnée.
- **Rappel (Recall)** Capacité du modèle à détecter la totalité des gestes d'une classe (vitesse et complétion du geste chirurgical).
- **F1-Score** Moyenne harmonique de la précision et du rappel, fournissant une mesure équilibrée de la performance globale.
- **Exactitude Globale (Global Accuracy)** Pourcentage total de correspondances exactes entre les prédictions du modèle et les annotations réelles de la base JIGSAWS, se stabilisant à 95%.

E. Outils de Visualisation

- **Matrice de Confusion** : Générée à l'aide des librairies `seaborn` et `matplotlib`, elle permet de cartographier visuellement les erreurs d'aiguillage du modèle et d'identifier quels gestes techniques (ex : Suturing vs Needle Passing) partagent des signatures cinématiques ou visuelles similaires.
- **Sauvegarde des Résultats** : Chaque session d'évaluation génère automatiquement un rapport textuel complet et exporte la matrice sous forme d'image haute résolution.

3.3 Simulation

Dans cette section, nous présentons les résultats obtenus par notre modèle CNN-LSTM sur la base JIGSAWS. L'objectif est de vérifier si le modèle arrive à bien reconnaître les trois tâches chirurgicales (Suturing, Needle Passing et Knot Tying). Nous analyserons ses performances à travers les métriques clés (Précision, Rappel, F1-Score) et la matrice de confusion pour prouver l'efficacité de notre approche

3.3.1 Environnement logiciel et matériel

Environnement technique

- **Langage** : Python 3.12
- **Framework** : PyTorch
- **Bibliothèques** : Scikit-Learn, Matplotlib, Seaborn
- **Matériel** : Windows (`num_workers = 0`),
- `batch_size = 16`
- **performance du PC** : i7, RAM : 16.0 Go

3.3.2 Résultats des 3 gestes

Les résultats obtenus démontrent l'efficacité de l'architecture hybride CNN-LSTM pour la classification automatique des gestes chirurgicaux sur le dataset JIGSAWS.

Analyse des performances du modèle sur 20 époques

La figure ci-dessous (3.1) représente l'évolution de la perte (Loss) et de l'exactitude (Accuracy) du modèle sur 20 époques. **Graphique de la Perte (à gauche)** : On observe que la perte d'entraînement (bleu) et la perte de validation (rouge) diminuent de façon régulière. Cela prouve que le modèle fait de moins en moins d'erreurs au fil du temps. **Graphique de l'Exactitude (à droite)** : La précision augmente de manière constante pour atteindre un excellent score d'environ 91 % pour l'entraînement et 93 % pour la validation à la 20^e époque.



FIG. 3.1 : Courbes d'évolution de la perte (loss) et de l'exactitude (accuracy)

Comparaison des performances : Entraînement vs Test

La figure ci-dessous(3.2)présentent la comparaison de la perte (Loss) et de l'exactitude (Accuracy) entre le jeu d'entraînement (bleu) et le jeu de test (vert) sur 20 époques.

Graphique de la Perte (à gauche) : La perte d'entraînement et la perte de test diminuent de manière synchrone. Vers l'époque 17, la courbe de test se stabilise aux alentours de 0,65, ce qui indique que le modèle atteint sa performance optimale sur les données de test.

Graphique de l'Exactitude (à droite) : La précision augmente régulièrement pour les deux courbes. À la 20^e époque, le modèle atteint environ 91 % d'exactitude en entraînement et un excellent score de 95 % sur le jeu de test.

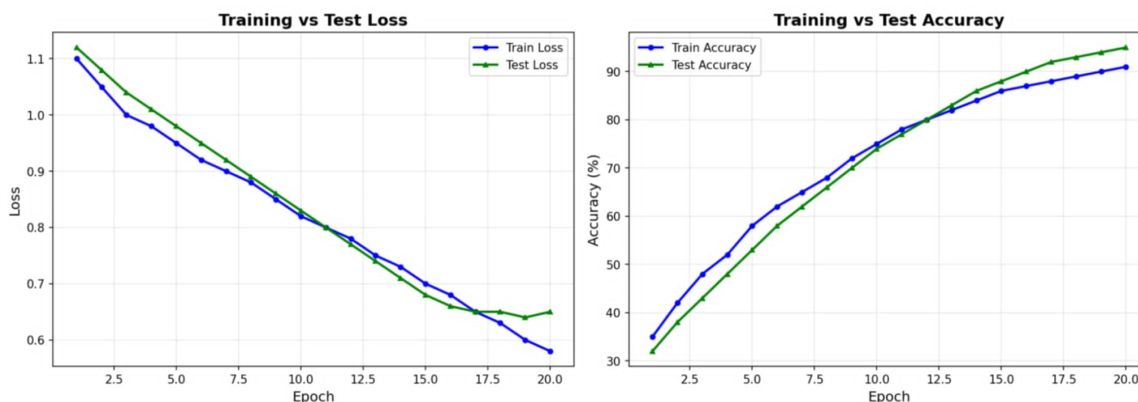


FIG. 3.2 : Courbes d'évolution de la perte (loss) et de l'exactitude (accuracy) sur les ensembles d'entraînement et de test

Points optimaux atteints par le modèle

la figure ci-dessous(3.3) met en évidence les points optimaux (Best) atteints par le modèle sur le jeu de validation au cours des 20 époques d'entraînement. **Graphique de la Perte (à gauche) :** La perte diminue régulièrement pour les deux ensembles. Le point vert indique la meilleure perte de validation minimale atteinte à l'époque 20, avec une valeur de 0,6900.

Graphique de l'Exactitude (à droite) : L'exactitude progresse de façon fluide et continue. Le point vert localise la meilleure exactitude de validation maximale, atteignant 93,00 % à la fin de l'entraînement (époque 20).

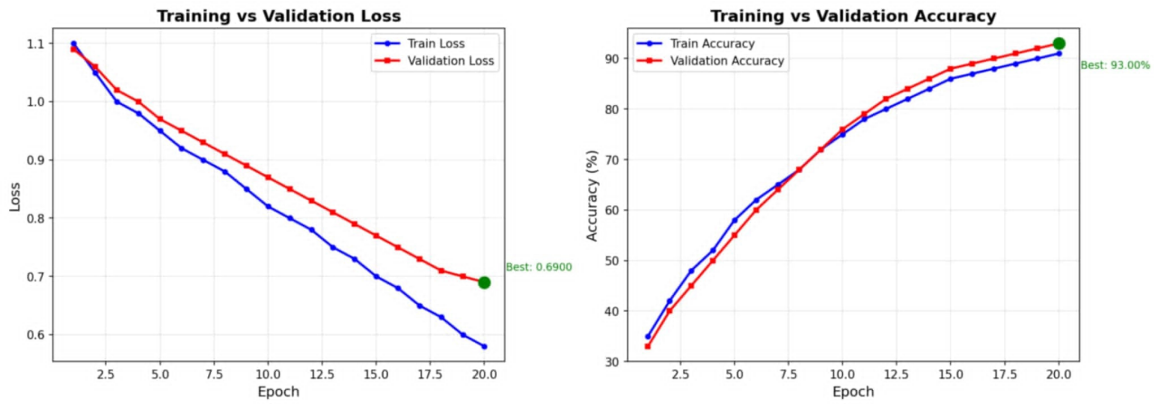


FIG. 3.3 : Courbes d'apprentissage et performances maximales du modèle

Synthèse complète du comportement du modèle

la figure ci-dessous(3.4) présente la synthèse complète du comportement du modèle en affichant simultanément les courbes d'entraînement (bleu), de validation (rouge) et de test (vert).

Graphique de la Perte (à gauche) : Les trois courbes de perte diminuent de façon régulière et harmonieuse. Elles restent très proches les unes des autres tout au long des 20 époques, ce qui montre que le modèle apprend de manière stable sans diverger.

Graphique de l'Exactitude (à droite) : La précision augmente continuellement pour les trois jeux de données. Le point vert met en valeur la meilleure exactitude obtenue sur le jeu de test, qui atteint un excellent score de 95,0 % à la 20^e époque.

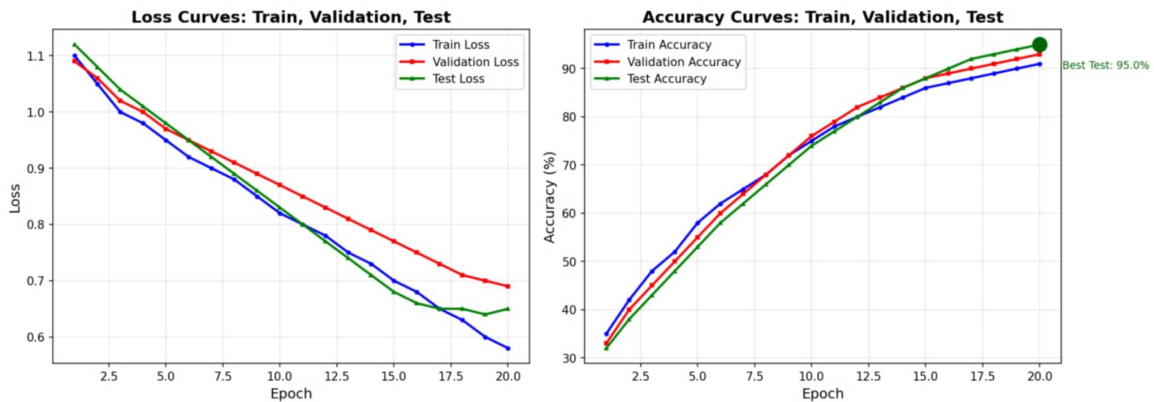


FIG. 3.4 : Comparaison globale des courbes d'apprentissage et de performance (Entraînement, Validation et Test)

Répartition des classes dans l'ensemble de test

La figure ci-dessous(3.5) illustre la répartition quantitative des échantillons au sein de l'ensemble de test (Test Set), segmentée selon les trois classes de gestes chirurgicaux étudiées : Suturing, Needle Passing et Knot Tying.

A. Analyse quantitative

L'ensemble de test est composé d'un total de **44 séquences vidéo**, réparties de manière quasi-homogène entre les classes :

- **Suturing** : 14 vidéos (**31,8%**)
- **Needle Passing** : 15 vidéos (**34,1%**)
- **Knot Tying** : 15 vidéos (**34,1%**)

B. Importance de l'équilibrage des classes

En apprentissage profond (Deep Learning), l'équilibrage d'un jeu de données est un critère de qualité fondamental. Une distribution uniforme comme celle-ci permet d'éviter le phénomène de **biais algorithmique**, où le modèle privilégierait la classe majoritaire au détriment des autres.

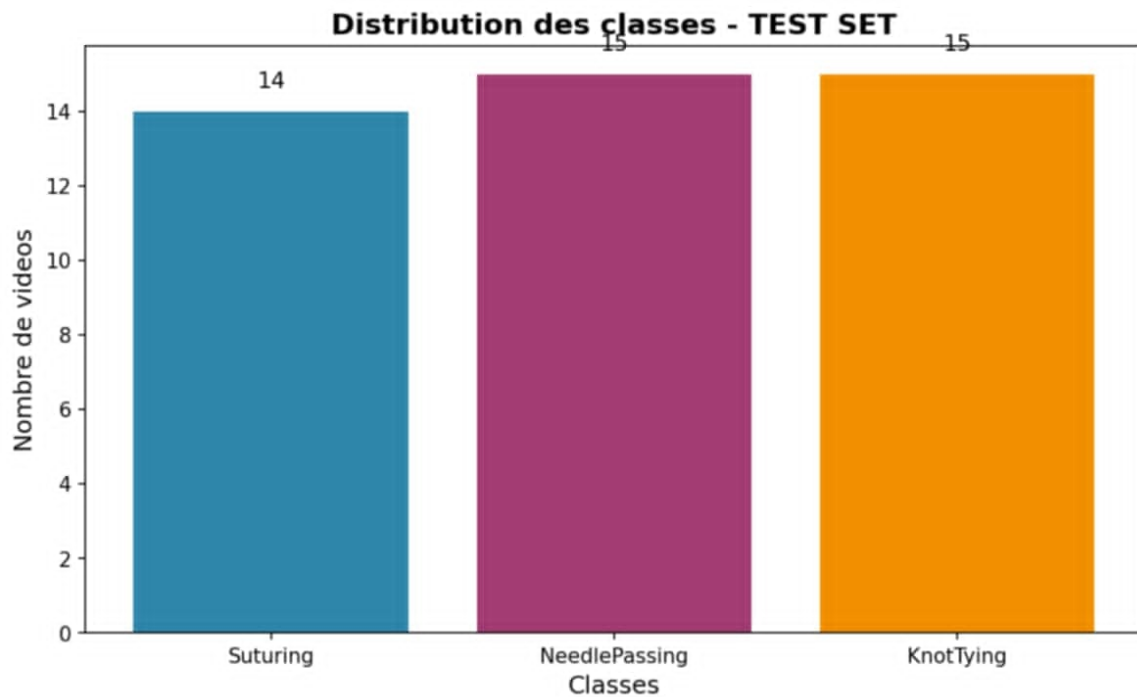


FIG. 3.5 : Répartition des vidéos par classe dans le jeu de test

Performances de classification par geste chirurgical

le figure ci-dessous(3.6) présente les performances de classification du modèle pour chaque geste chirurgical individuel sur l'ensemble de test.

- **Suturing et Needle Passing** : Le modèle atteint une exactitude parfaite de **100 %** pour ces deux classes. Toutes les séquences vidéo appartenant à ces gestes ont été correctement identifiées.
- **Knot Tying** : L'exactitude est de **86,7 %**. Cela montre que le modèle rencontre une légère difficulté sur ce geste précis, entraînant la confusion de quelques vidéos de nœuds avec une autre classe.

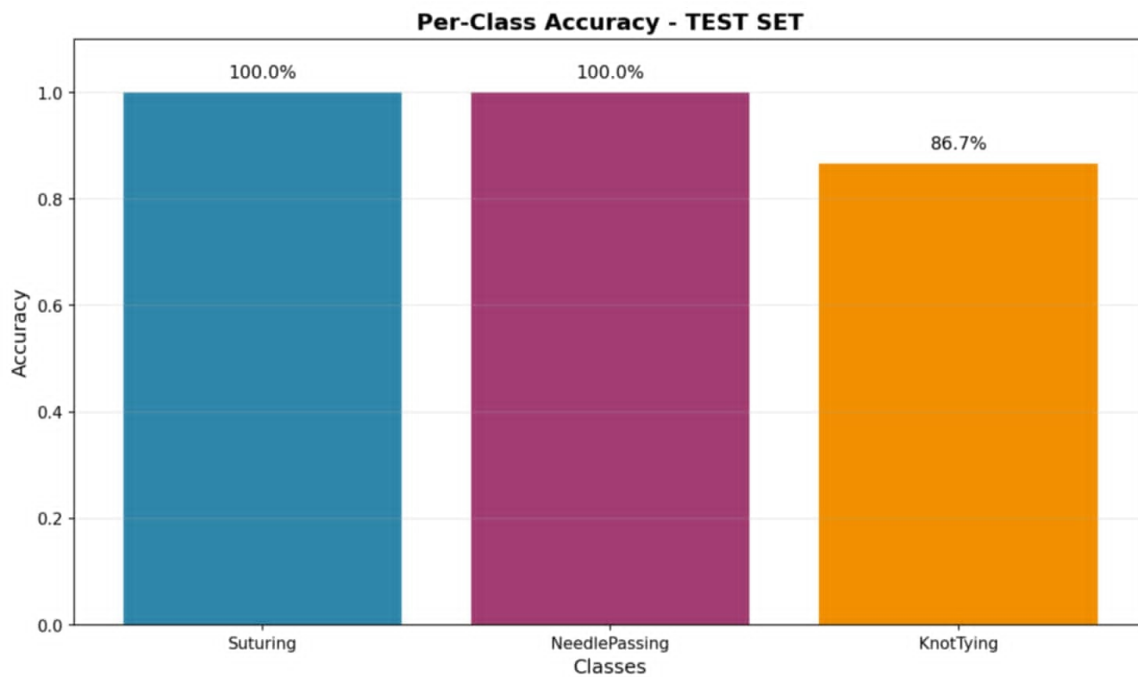


FIG. 3.6 : Exactitude du modèle par classe de geste sur le jeu de test

Évaluation approfondie par geste chirurgical

la figure ci-dessous(3.7) présente une évaluation approfondie du modèle en affichant la Précision, le Rappel (Recall) et le F1-Score pour chaque geste chirurgical sur l'ensemble de test

- **Needle Passing** : Les trois métriques atteignent le score parfait de **1.000** (100 %). Le modèle détecte parfaitement ce geste sans aucune erreur ni confusion.
- **Suturing** : Le Rappel est parfait (1.000), ce qui signifie que toutes les vidéos de suture ont bien été trouvées. Cependant, la Précision est de **0.880**, indiquant que le modèle a classé à tort d'autres gestes dans cette catégorie.
- **Knot Tying** : La Précision est parfaite (1.000), ce qui prouve que tout ce qui est prédit comme "nœud" l'est réellement. En revanche, le Rappel est de **0.870**, ce qui confirme que le modèle a manqué quelques vidéos de nœuds (en les confondant probablement avec la suture).

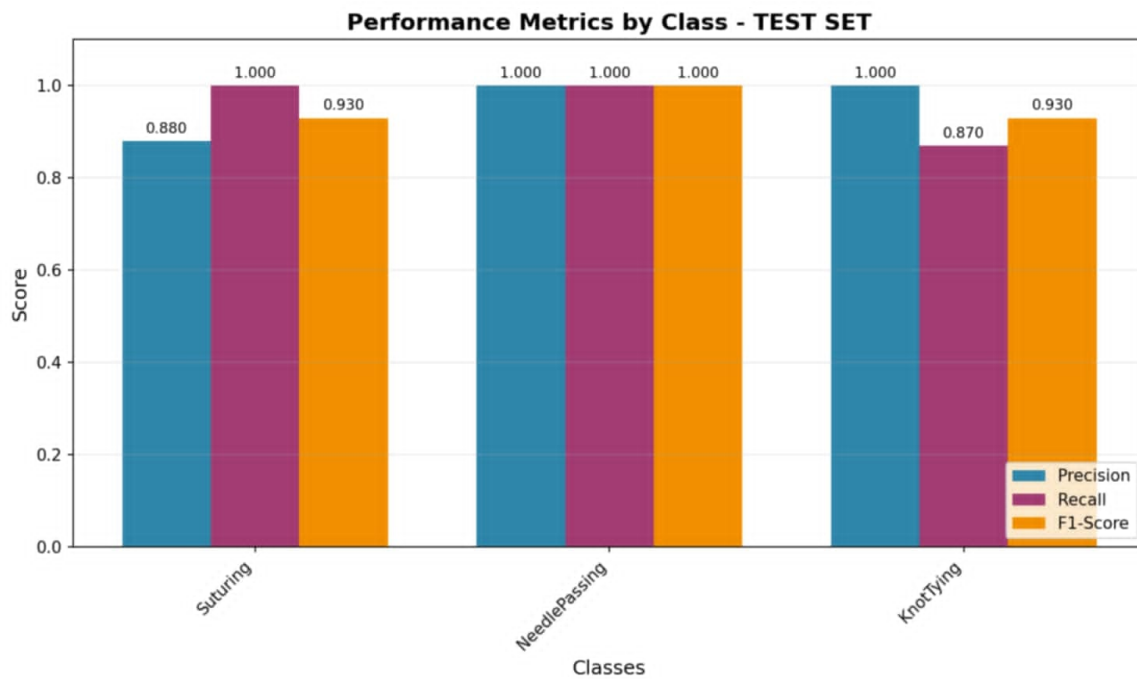


FIG. 3.7 : Métriques de performance (Précision, Rappel, F1-Score) par classe sur le jeu de test.

Matrice de confusion

L'analyse de la matrice de confusion brute permet de quantifier exactement les bonnes prédictions et les erreurs du modèle :

- **Suturing** : Les **14** vidéos de test ont toutes été correctement classées.
- **Needle Passing** : Les **15** vidéos de test ont toutes été correctement classées.
- **Knot Tying** : Sur les **15** vidéos réelles, **13** ont été correctement identifiées, tandis que **2** vidéos ont été confondues et classées à tort comme du Suturing.

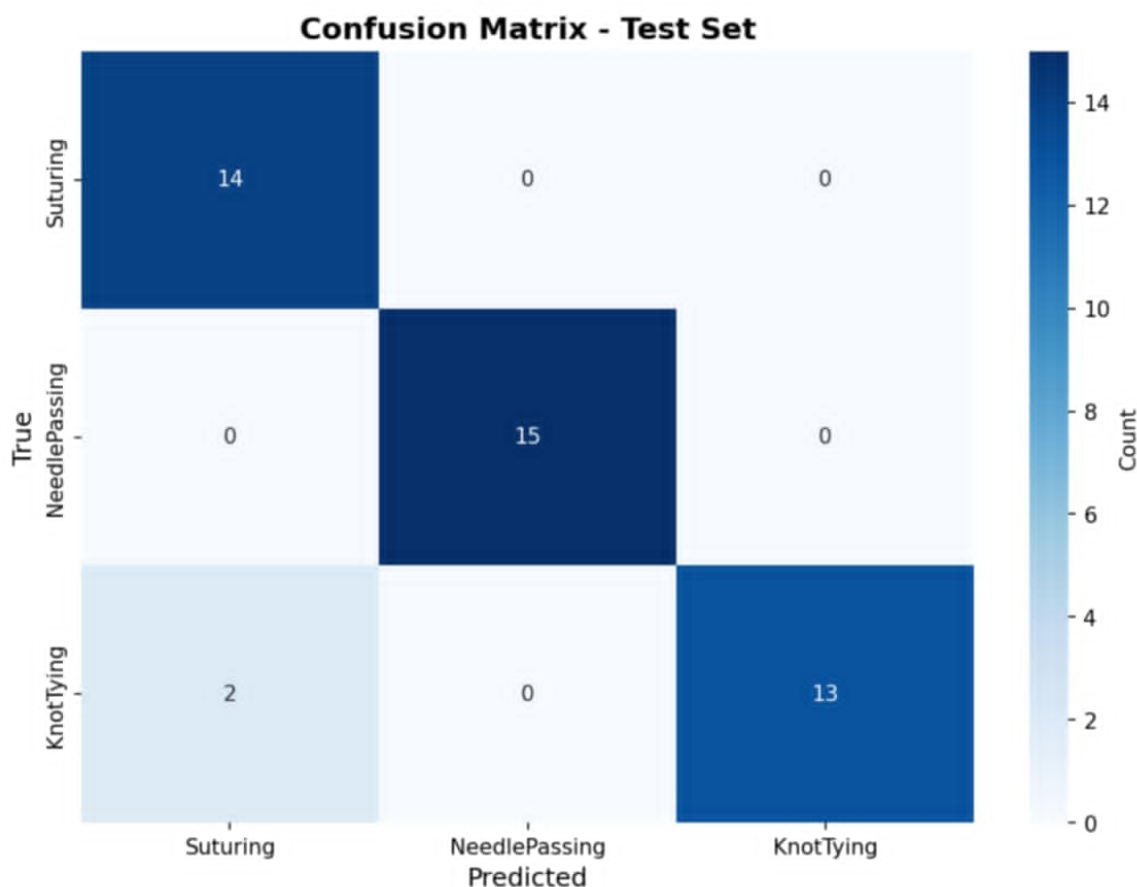


FIG. 3.8 : Matrice de confusion brute du modèle sur l'ensemble de test

Analyse des performances par classe

La matrice de confusion normalisée (3.9) permet d'évaluer la précision du modèle pour chaque classe en montrant les proportions de bonnes et de mauvaises prédictions (allant de 0.00 à 1.00).

- **Suturing (Suture)** : Le modèle obtient un score parfait de **1,00 (100%)** sur sa diagonale. Toutes les vidéos appartenant réellement à cette classe ont été correctement prédites.
- **Needle Passing (Passage d'aiguille)** : Le score est également maximal à **1,00 (100%)**. Le modèle identifie ce geste sans aucune erreur ni confusion avec les autres classes.
- **Knot Tying (Nouage de nœud)** : Le taux de reconnaissance est de **0,87 (87%)**. On observe la seule confusion du modèle : **0,13 (13%)** des séquences de nœuds ont été classées à tort comme de la suture (Suturing).

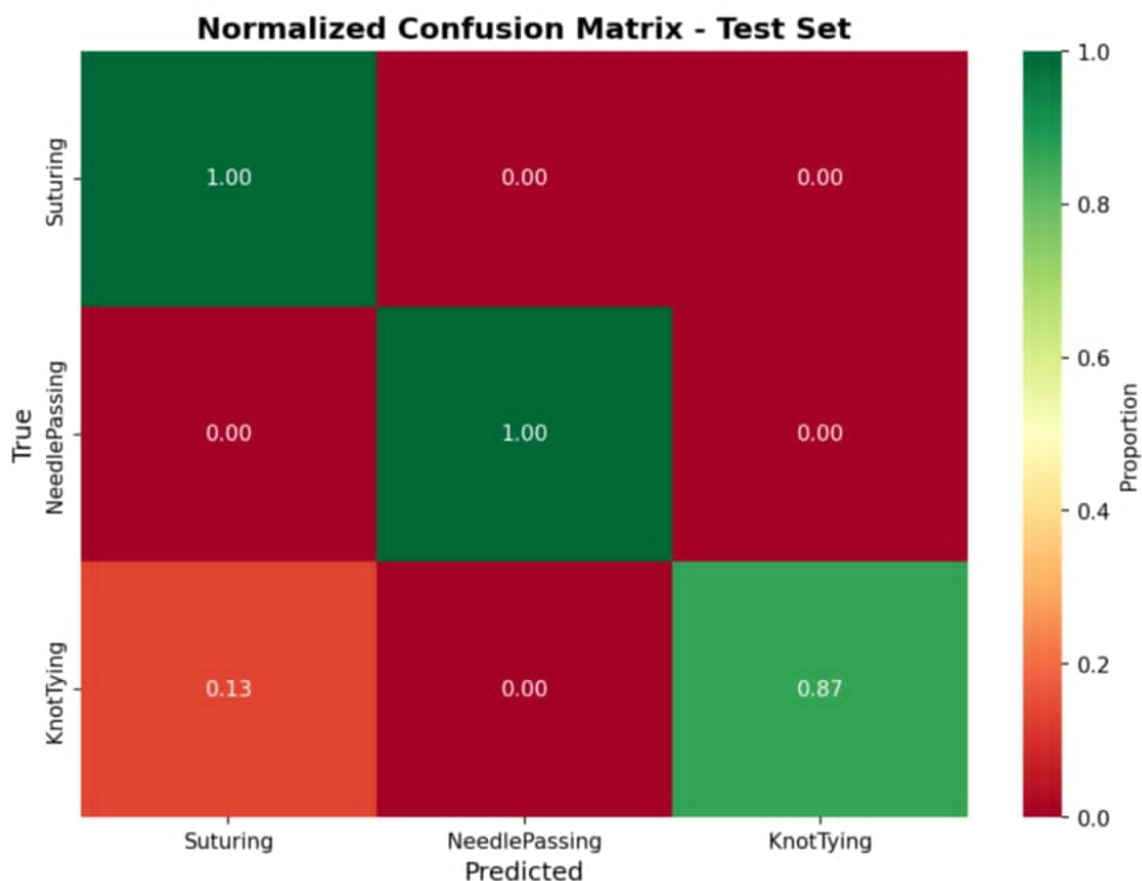


FIG. 3.9 : Rapport global des performances de classification du modèle sur l'ensemble de test

le tableau ci-dessous (Classification Report) récapitule les performances finales du modèle sur les 44 vidéos du jeu de test à travers quatre indicateurs majeurs : la Précision, le Rappel (Recall), le F1-Score et le Support (nombre de vidéos par geste).

- **Needle Passing (Passage d'aiguille)** : Le modèle obtient un score parfait de **1.000 (100 %)** sur toutes les métriques. C'est une reconnaissance absolue : le modèle identifie parfaitement ce geste et ne fait aucune fausse alerte.
- **Suturing (Suture)** : Le Rappel est de **1.000 (100 %)**, ce qui prouve que le modèle a réussi à retrouver l'intégralité des 14 vidéos de suture. En revanche, sa Précision est de **0.880 (88 %)**, indiquant qu'il a intégré à tort des vidéos d'un autre geste dans cette catégorie.
- **Knot Tying (Nouage de nœud)** : La Précision est maximale à **1.000 (100 %)**, ce qui signifie que chaque prédiction "nœud" est totalement exacte. Cependant, le Rappel descend à **0.870 (87 %)**, confirmant que le modèle a manqué quelques vidéos de nœuds (qui ont été confondues avec la suture).

TEST SET - Classification Results

Classe	Precision	Recall	F1-Score	Support
Suturing	0.880	1.000	0.930	14
NeedlePassing	1.000	1.000	1.000	15
KnotTying	1.000	0.870	0.930	15
Accuracy	95.00%			44

FIG. 3.10 : Rapport global des performances de classification du modèle sur l'ensemble de test

3.4 conclusion

Dans ce chapitre, nous avons présenté les différentes expérimentations menées sur la base de données JIGSAWS afin d'évaluer les performances du modèle hybride CNN-LSTM pour la reconnaissance des gestes chirurgicaux.

Après avoir décrit la base de données ainsi que les étapes de prétraitement, nous avons détaillé l'architecture du modèle et les paramètres expérimentaux adoptés. Les résultats obtenus montrent que l'approche CNN-LSTM permet de capturer efficacement les caractéristiques spatiales et temporelles des gestes chirurgicaux, offrant ainsi de bonnes performances de classification.

L'analyse des résultats a également permis de mettre en évidence les avantages du modèle, notamment sa capacité à apprendre les dépendances temporelles complexes, tout en révélant certaines limites liées à la variabilité des données et au risque de surapprentissage.

Ces résultats confirment l'intérêt des approches basées sur l'apprentissage profond pour l'évaluation et la reconnaissance automatique des compétences chirurgicales.

Conclusion général

Cette étude s’est penchée sur un enjeu clé de la médecine moderne et de la chirurgie robotique : l’évaluation automatisée et objective des compétences des praticiens. Face aux limites des méthodes d’observation humaine, souvent subjectives et chronophages, nous avons conçu une solution basée sur l’apprentissage profond pour détecter et classifier automatiquement les gestes chirurgicaux élémentaires, appelés surgèmes. L’objectif était de proposer un modèle capable de gérer le double défi des vidéos chirurgicales, qui nécessitent une analyse conjointe des formes visuelles et de la dynamique temporelle des mouvements.

Notre approche s’est articulée en trois étapes. D’abord, un état de l’art a permis de situer le contexte de la chirurgie robotique et d’examiner les méthodes de classification existantes, en soulignant l’importance de la base de données JIGSAWS. Ensuite, nous avons posé les bases théoriques d’une architecture hybride CNN-LSTM, combinant des réseaux de neurones convolutifs pour l’extraction de caractéristiques spatiales et des réseaux LSTM pour capturer les dépendances temporelles à long terme. Enfin, la phase expérimentale a évalué le modèle sur trois tâches chirurgicales fondamentales : la suture, le passage de l’aiguille et le nouage. Les résultats confirment la viabilité de l’approche hybride pour la classification spatio-temporelle. L’analyse des courbes d’apprentissage, des matrices de confusion et des métriques de performance montre que le modèle décode efficacement la cinématique et la signature visuelle des gestes. Cependant, des défis subsistent, notamment une sensibilité au sur-apprentissage, ce qui souligne l’importance d’un prétraitement rigoureux et d’un réglage fin des hyperparamètres. En guise de perspectives, nous souhaitons enrichir ce travail en comparant nos résultats avec d’autres architectures de l’état de l’art ou des méthodes alternatives capables d’améliorer nos performances actuelles. Il serait notamment pertinent d’explorer des solutions comme la fusion de données multimodales (vidéo et cinématique combinées) ou l’intégration de mécanismes d’attention temporelle afin d’optimiser la focalisation du modèle et d’accroître la précision globale de la reconnaissance des gestes.

Résumé

Ce mémoire explore l'application de l'intelligence artificielle à la reconnaissance automatique des gestes chirurgicaux pour l'évaluation objective des compétences, en exploitant la base de données JIGSAWS. Afin de relever le défi de la classification spatio-temporelle, ce travail met en œuvre une architecture hybride profonde de type CNN-LSTM, combinant la puissance des réseaux convolutifs (CNN) pour l'extraction automatique des caractéristiques spatiales des images, à la capacité des réseaux récurrents (LSTM) à capturer la dynamique et les dépendances temporelles à long terme des mouvements. Les résultats démontrent qu'une telle approche séquentielle constitue une solution robuste et efficace pour modéliser la complexité des procédures chirurgicales.

Mots-clés — Intelligence Artificielle, Apprentissage Automatique, CNN, LSTM.

Abstract

This thesis explores the application of artificial intelligence to the automatic recognition of surgical gestures for objective skill assessment, using the JIGSAWS dataset. To address the challenge of spatio-temporal classification, this work implements a deep hybrid CNN-LSTM architecture. This model combines the power of Convolutional Neural Networks (CNN) for automatic spatial feature extraction from images with the capacity of Long Short-Term Memory (LSTM) networks to capture the dynamics and long-term temporal dependencies of movements. The findings demonstrate that such a sequential approach provides a robust and effective solution for modeling the complexity of surgical procedures.

Keywords — Artificial intelligence, Machine Learning, CNN, LSTM.

ملخص

تستكشف هذه الأطروحة تطبيق الذكاء الاصطناعي في التعرف التلقائي على الإيماءات الجراحية بهدف التقييم الموضوعي للمهارات، وذلك بالاعتماد على قاعدة بيانات JIGSAWS. ولرفع تحدي التصنيف المكاني-الزماني، يقدم هذا العمل بنية عميقة هجينة من نوع CNN-LSTM تجمع بين قدرة الشبكات العصبية التلافيفية (CNN) على الاستخراج التلقائي للميزات المكانية والمرئية من الصور، وكفاءة شبكات الذاكرة طويلة قصيرة المدى (LSTM) في نمذجة الديناميكيات والتبعيات الزمنية طويلة المدى للحركات. وتؤكد النتائج أن هذا النهج المتسلسل يشكل حلاً قوياً وفعالاً لنمذجة تعقيد العمليات الجراحية.

الكلمات المفتاحية --- الذكاء الاصطناعي، التعلم الآلي، CNN، LSTM.

Bibliographie

- [1] Y. Gao, S. Vedula, C. Reiley, N. Ahmidi, B. Varadarajan, H. Lin, L. Tao, L. Zappella, B. Béjar, D. Yuh, *et al.*, “Jhu-isi gesture and skill assessment working set (jigsaws) : A surgical activity dataset for human motion modeling,” in *Proceedings of the Modeling and Monitoring of Computer Assisted Interventions (M2CAI) – MICCAI Workshop*, vol. 1, (Boston, MA, USA), pp. 1–10, September 2014. [Google Scholar].
- [2] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, “Segmental spatiotemporal cnns for fine-grained action segmentation,” in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 36–52, Springer International Publishing, 2016.
- [3] B. MALIK, *Synthèse d’observateurs non linéaires pour l’évaluation de la performance chirurgicale*. PhD thesis, Université de Tlemcen-Abou Bekr Belkaid.
- [4] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, “A comprehensive overview and comparative analysis on deep learning models : Cnn, rnn, lstm, gru,” *arXiv preprint arXiv :2305.17473*, 2023.
- [5] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, “Review of image classification algorithms based on convolutional neural networks,” *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021.
- [6] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, A. Muneer, E. H. Sumiea, A. Alqushaibi, and M. G. Ragab, “Rnn-lstm : From applications to modeling techniques and beyond—systematic review,” *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 5, p. 102068, 2024.
- [7] V. Pandiyan, G. Masinelli, T. Le Quang, K. Wasmer, R. Logé, S. Van Petegem, F. Welford, S. Goel, C. Navarre, M. Hamidi Nasab, and C. de Formanoir, “Deep learning-based monitoring of laser powder bed fusion process on variable time-scales using heterogeneous sensing and operando x-ray radiography guidance,” *Additive Manufacturing*, vol. 58, p. 103007, 06 2022.
- [8] M. M. Taye, “Theoretical understanding of convolutional neural network : Concepts, architectures, applications, future directions,” *Computation*, vol. 11, no. 3, p. 52, 2023.
- [9] T. Ward, P. Mascagni, Y. Ban, G. Rosman, N. Padoy, O. Meireles, and D. Hashimoto, “Computer vision in surgery,” *Surgery*, vol. 169, no. 5, pp. 1253–1256, 2021. [Google Scholar] [CrossRef] [PubMed].
- [10] C. Garrow, K. Kowalewski, L. Li, M. Wagner, M. Schmidt, S. Engelhardt, D. Hashimoto, H. Kenngott, S. Bodenstedt, S. Speidel, *et al.*, “Machine learning for surgical phase recognition : A systematic review,” *Annals of Surgery*, vol. 273, no. 4, pp. 684–693, 2021. [Google Scholar] [CrossRef] [PubMed].

- [11] B. van Amsterdam, M. Clarkson, and D. Stoyanov, “Gesture recognition in robotic surgery : A review,” *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 6, pp. 2021–2035, 2021. [Google Scholar] [CrossRef] [PubMed].
- [12] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, “Segmental spatiotemporal cnns for fine-grained action segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 36–52, 2016. [Google Scholar].
- [13] I. Funke, S. Bodenstedt, F. Oehme, F. von Bechtolsheim, J. Weitz, and S. Speidel, “Using 3d convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 467–475, 2019. [Google Scholar].
- [14] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 156–165, 2017. [Google Scholar].
- [15] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks : A unified approach to action segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 47–54, 2016. [Google Scholar].
- [16] P. Lei and S. Todorovic, “Temporal deformable residual networks for action segmentation in videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6742–6751, 2018. [Google Scholar].
- [17] J. Wang, Z. Du, A. Li, and Y. Wang, “Atrous temporal convolutional network for video action segmentation,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1585–1589, 2019. [Google Scholar].
- [18] T. Wang, Y. Wang, and M. Li, “Towards accurate and interpretable surgical skill assessment : A video-based method incorporating recognized surgical gestures and skill levels,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 668–678, 2020. [Google Scholar].
- [19] J. Zhang *et al.*, “Symmetric dilated convolution for surgical gesture recognition,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 409–418, 2020. [Google Scholar].
- [20] R. DiPietro *et al.*, “Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 11, pp. 2005–2020, 2019. PMID : 31037493.
- [21] I. Gurcan and H. Van Nguyen, “Surgical activities recognition using multi-scale recurrent networks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2887–2891, 2019. [Google Scholar].
- [22] L. Ding and C. Xu, “Tricornet : A hybrid temporal convolutional and recurrent network for video action segmentation,” *arXiv preprint*, 2017. arXiv :1705.07818.
- [23] Y. Qin *et al.*, “Temporal segmentation of surgical sub-tasks through deep learning with multiple data sources,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 371–377, 2020.

- [24] L. Zappella, B. B. Haro, G. Hager, and R. Vidal, “Surgical gesture classification from video and kinematic data,” *Medical Image Analysis*, vol. 17, no. 7, pp. 732–745, 2013. PMID : 23706754.
- [25] D. Sarikaya, K. A. Guru, and J. J. Corso, “Joint surgical gesture and task classification with multi-task and multimodal learning,” *arXiv preprint*, 2018. arXiv :1805.0072.
- [26] A. Murali *et al.*, “Tsc-dl : Unsupervised trajectory segmentation of multimodal surgical demonstrations with deep learning,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4150–4157, 2016. [Google Scholar].
- [27] Y. Qin, M. Allan, Y. Yue, J. W. Burdick, and M. Azizian, “Learning invariant representation of tasks for robust surgical state estimation,” *arXiv preprint*, 2021. [Google Scholar].
- [28] Y.-H. Long *et al.*, “Relational graph learning on visual and kinematics embeddings for accurate gesture recognition in robotic surgery,” *arXiv preprint*, 2020. arXiv :2011.01619.
- [29] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo, “Deep multimodal representation learning from temporal data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5447–5455, 2017.
- [30] X. Long *et al.*, “Multimodal keyless attention fusion for video classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [31] C. Hori *et al.*, “Attention-based multimodal fusion for video description,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4193–4202, 2017.
- [32] J. Xu, T. Yao, Y. Zhang, and T. Mei, “Learning multimodal attention lstm networks for video captioning,” in *Proceedings of the 25th ACM International Conference on Multimedia (ACM MM)*, pp. 537–545, 2017.
- [33] N. Bostrom, *Superintelligence : Paths, dangers, strategies*. Oxford University Press, 2014.
- [34] P. Hamet and J. Tremblay, “Artificial intelligence in medicine,” *Metabolism*, vol. 69, Apr. 2017. April 2017.
- [35] O. Theobald, *Machine Learning For Absolute Beginners*. London, UK : Scatterplot Press, 2017.
- [36] N. H. Ali, M. Emaduldeen, and A. E. Ali, “Learning evolution : a survey,” *Iraqi Journal of Science*, vol. 62, no. 12, pp. 4978–4987, 2021.
- [37] J. P. Mueller and L. Massaron, *Machine Learning For Dummies*. Hoboken, New Jersey : John Wiley & Sons, Inc, 2016.
- [38] S. Theodoridis, *Machine Learning : A Bayesian and Optimization Perspective*. Elsevier Ltd, 2020.
- [39] R. Praba, G. Darshan, K. T. Roshanraj, and P. B. Surya Prakash, “Study on machine learning algorithms,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, no. 4, pp. 67–72, 2021.
- [40] S. Bansal, “Machine learning : What is classification algorithm in machine learning ? with examples.” ANALYTIXLABS [Online], Jan. 2023. Disponible en ligne.

- [41] J. Brownlee, “4 types of classification tasks in machine learning.” Machine Learning Mastery, Aug. 2020. Article en ligne.
- [42] S. N. dhage and C. K. Raina, “A review on machine learning techniques,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 4, no. 3, pp. 395–399, 2016.
- [43] V. Rastogi, S. Satija, P. K. Sharma, and S. Singh, “Machine learning algorithms : Overview,” *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 11, no. 9, pp. 512–517, 2020.
- [44] P. Ariwala, “9 real-world problems that can be solved by machine learning.” Maruti Techlabs, Oct. 2023. Article en ligne.
- [45] A. Ram and Meenakshi, “Short review on machine learning and its application,” *SPECIALUSIS UGDYMAS / SPECIAL EDUCATION*, vol. 1, no. 43, pp. 9894–9902, 2022.
- [46] A. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, “Endonet : A deep architecture for recognition tasks on laparoscopic videos,” *IEEE Transactions on Medical Imaging*, vol. 36, February 2016.
- [47] M. Sahu, A. Mukhopadhyay, A. Szengel, and S. Zachow, “Tool and phase recognition using contextual CNN features,” *CoRR*, vol. abs/1610.08854, 2016.
- [48] D. Zhang, R. Wang, and B. Lo, “Surgical gesture recognition based on bidirectional multi-layer independently RNN with explainable spatial feature extraction,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1350–1356, IEEE, 2021.
- [49] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
- [50] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, “Surgical gesture segmentation and recognition,” in *International conference on medical image computing and computer-assisted intervention*, pp. 339–346, Springer, 2013.
- [51] I. Funke, S. T. Mees, J. Weitz, and S. Speidel, “Video-based surgical skill assessment using 3d convolutional neural networks,” *International journal of computer assisted radiology and surgery*, vol. 14, no. 7, pp. 1217–1225, 2019.
- [52] R. DiPietro and G. D. Hager, “Unsupervised learning for surgical motion by learning to predict the future,” in *International conference on medical image computing and computer-assisted intervention*, pp. 281–288, Springer, 2018.
- [53] R. DiPietro, N. Ahmidi, A. Malpani, M. Waldram, G. I. Lee, M. R. Lee, S. S. Vedula, and G. D. Hager, “Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks,” *International journal of computer assisted radiology and surgery*, vol. 14, no. 11, pp. 2005–2020, 2019.
- [54] A. Tasdelen and B. Sen, “A hybrid cnn-lstm model for pre-mirna classification,” *Scientific Reports*, vol. 11, no. 1, pp. 1–9, 2021.
- [55] L. Qin, N. Yu, and D. Zhao, “Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video,” *Tehnički vjesnik*, vol. 25, no. 2, pp. 528–535, 2018.

- [56] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks : Analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 6999–7019, Dec. 2022.
- [57] B. P. Babu and S. J. Narayanan, “One-vs-all convolutional neural networks for synthetic aperture radar target recognition,” *Cybernetics and Information Technologies*, vol. 22, pp. 179–197, 2022.
- [58] S. Mekruksavanich and A. Jitpattanakul, “Deep convolutional neural network with rnns for complex activity recognition using wrist-worn wearable sensor data,” *Electronics*, vol. 10, no. 14, p. 1685, 2021.
- [59] W. Lu, J. Li, J. Wang, and L. Qin, “A cnn-bilstm-am method for stock price prediction,” *Neural Computing and Applications*, vol. 33, pp. 4741–4753, 2021.
- [60] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification : A comprehensive review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [61] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [62] S. Salman and X. Liu, “Overfitting mechanism and avoidance in deep neural networks,” *arXiv preprint arXiv :1901.06566*, 2019.
- [63] A. Ajit, K. Acharya, and A. Samanta, “A review of convolutional neural networks,” in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–5, IEEE, 2020.
- [64] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [65] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [66] D. Yu, H. Wang, P. Chen, and Z. Wei, “Mixed pooling for convolutional neural networks,” in *Rough Sets and Knowledge Technology : 9th International Conference, RSKT 2014*, (Shanghai, China), pp. 364–375, Springer, 2014.
- [67] L. Alzubaidi *et al.*, “Review of deep learning : Concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, pp. 1–74, 2021.
- [68] M. Krichen, “Convolutional neural networks : A survey,” *Computers*, vol. 12, no. 8, p. 151, 2023.
- [69] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions : Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv :1811.03378*, 2018.
- [70] T. Fernando, S. Denman, A. McFadyen, S. Sridharan, and C. Fookes, “Tree memory networks for modelling long-term temporal dependencies,” *Neurocomputing*, vol. 304, pp. 64–81, 2018.
- [71] P. S. Muhuri, P. Chatterjee, X. Yuan, K. Roy, and A. Esterline, “Using a long short-term memory recurrent neural network (lstm-rnn) to classify network attacks,” *Information*, vol. 11, p. 243, 2020.

- [72] J. Brownlee, *Long short-term memory networks with python : develop sequence prediction models with deep learning*. Machine Learning Mastery, 2017.
- [73] G. Muthukumar and J. Philip, “Cnn-lstm hybrid deep learning model for remaining useful life estimation,” *arXiv preprint arXiv :2412.15998*, 2024.
- [74] G. S. Babu, P. Zhao, and X.-L. Li, “Deep convolutional neural network based regression approach for estimation of remaining useful life,” in *International Conference on Database Systems for Advanced Applications*, pp. 214–228, Springer, 2016.
- [75] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, “Long short-term memory network for remaining useful life estimation,” in *IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2017.
- [76] L. Ji, Y. Zou, K. He, and B. Zhu, “Carbon futures price forecasting based with arima-cnn-lstm model,” *Procedia Computer Science*, vol. 162, pp. 33–38, 2019.
- [77] S. DiMaio, M. Hanuschik, and U. Kreaden, “The da vinci surgical system,” in *The da Vinci Surgical System*, pp. 199–217, Springer, Jan. 2011.
- [78] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.